AFRL-IF-RS-TR-2000-122
Final Technical Report
August 2000

PORTABLE COUPLED FIELD CAD

Coyote Systems

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. E117

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

DTIC QUALITY INSPECTED 4

20000925 161

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2000-122 has been reviewed and is approved for publication.

APPROVED: Clare D. Thiem

CLAIRE D. THIEM
Project Engineer

FOR THE DIRECTOR: [illegible]

NORTHRUP FOWLER, Technical Advisor
Information Technology Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify Air Force Research Laboratory/IFTC, 26 Electronic Pky, Rome, NY 13441-4514. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

PORTABLE COUPLED FIELD CAD

Per Ljung

Contractor:	Coyote Systems
Contract Number:	F30602-96-2-0305
Effective Date of Contract:	20 November 1996
Contract Expiration Date:	22 September 1999
Short Title of Work:	Portable Coupled Field Cad
Period of Work Covered:	Nov 96 – Sep 99
Principal Investigator:	Per Ljung
Phone:	(415) 346-4223
AFRL Project Engineer:	Clare D. Thiem
Phone:	(315) 330-4893

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Clare D. Thiem, AFRL/IFTC, 26 Electronic Pky, Rome, NY.

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE AUGUST 2000	3. REPORT TYPE AND DATES COVERED Final Nov 96 - Sep 99

4. TITLE AND SUBTITLE PORTABLE COUPLED FIELD CAD	5. FUNDING NUMBERS C - F30602-96-2-0305 PE - 63739E PR - E117 TA - 00 WU - 04
6. AUTHOR(S) Per Ljung	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Coyote Systems 2740 Van Ness Avenue #210 San Francisco CA 94109	8. PERFORMING ORGANIZATION REPORT NUMBER N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency 3701 North Fairfax Drive Arlington VA 22203-1714 Air Force Research Laboratory/IFTC 26 Electronic Pky Rome NY 13441-4514	10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2000-122

11. SUPPLEMENTARY NOTES
Air Force Research Laboratory Project Engineer: Clare D. Thiem/IFTC/(315) 330-4893

12a. DISTRIBUTION AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Coyote Systems has developed a coupled CAD tool for Microelectromechanical Systems (MEMS), called AutoMEMS, which accurately simulates the coupled fields in large, realistic MEMS devices with minimal user input. AutoMEMS generates a three dimensional (3D) solid model from a user-supplied two dimensional device layout. A device mesh is then created from the 3D solid model which can be automatically and adaptively refined for accurate results. Simulations are then conducted utilizing a fast Boundary Element Method (BEM). The resulting cross -capacitances and electrostatic forces can be automatically extracted. The final version of AutoMEMS was developed in a C++ environment using gnu-style tools and high-speed portable OpenGL graphics which allows it to run on Linux and Unix platforms. Two particular examples used to demonstrate the functionality of AutoMEMS were of a micromirror and Analog Devices ADXL76 accelerometer. Thus, an effective and easy-to-use MEMS simulation tool was developed and demonstrated.

14. SUBJECT TERMS Computer-Aided Design (CAD), Microelectromechanical Systems (MEMS), Boundary Element Method (BEM), Adaptive Meshing, Electromechanical, Electrostatic, Coupled Fields	15. NUMBER OF PAGES 136
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

Table of Contents

List of Figures

List of Tables

1.0 Executive Summary

1.1 Purpose of This Project

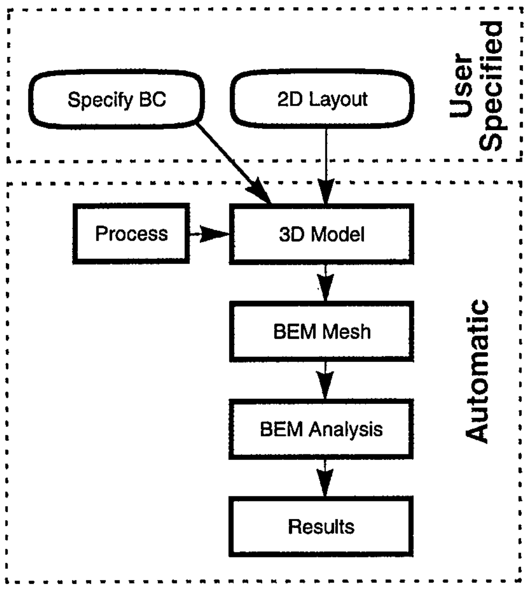
Coyote Systems has developed a coupled CAD tool for MEMS, called AutoMEMS®, which accurately simulates the coupled fields in large, realistic MEMS devices. AutoMEMS automatically creates 3D models from 2D MEMS layouts and accurately solves the coupled 3D partial differential equations using a fast boundary element method. Most simulation steps (e.g. 3D model generation, meshing, accuracy control) are completely automated, substantially eliminating the need for MEMS simulation experts.

Coyote has delivered AutoMEMS software to Beta Partners who have successfully used it to analyze their existing MEMS designs. Analog Devices for example has simulated the ADXL76 accelerometer in less than 1 minute.

1.2 Solving the Problem

Coyote Systems' approach is different from competing efforts because AutoMEMS simulations are designed to be performed with a minimal amount of user input. Figure 1 illustrates the data flow, with only the 2D design masks and boundary conditions required to be specified by the user.

FIGURE 1. Data flow to simulate a MEMS device.



To achieve this level of automation Coyote's AutoMEMS CAD tool:

- Automatically generates a 3D solid model from a user-supplied 2D device layout
- Automatically creates a device mesh from a 3D solid model
- Automatically and adaptively refines the mesh for accurate results
- Simulates PDEs using the fast Boundary Element Method (BEM)
- Automatically extracts cross-capacitances and electrostatic forces

Coyote Systems has developed, demonstrated and delivered all key technologies required to achieve the automation needed for an effective and easy-to-use MEMS simulation tool.

1.3 Sample MEMS Simulation

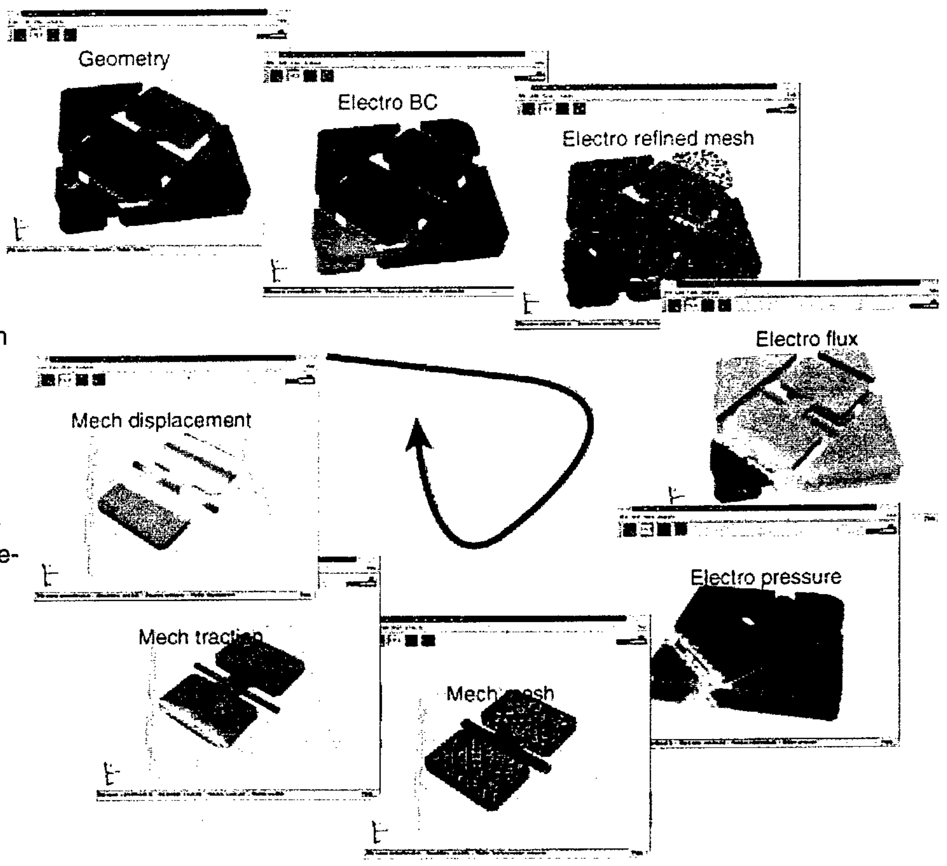
Coyote successfully implemented its unique numerical approach to solving MEMS design and simulation problems as stated above. With this framework and functionality, user interaction is minimized.

For example, we will show the steps to simulate the coupled electrostatic/mechanical behavior of a MEMS micromirror comparable to Texas Instruments's digital micromirror device (DMD). To perform a MEMS simulation, the user must provide a 2D layout of the device, process description and define the boundary conditions.

FIGURE 2. Coupled electrostatic-mechanical simulation of TI micromirror

Automated iterations

1. Import Geometry
2. Define Electro BC
3. Define Mech BC
4. Refine Electro Mesh
5. Solve Electro flux
6. Calc Electro pressure
7. Refine Mech mesh
8. Apply Mech traction
9. Solve Mech displacement
10. Done!



From the 2D layout masks, a 3D solid model is automatically generated by emulating common MEMS processing steps. Coyote Systems software also automatically discretizes the surfaces of the 3D solid model, creating a Boundary Element Method (BEM) mesh. The BEM mesh has continuous BEM elements with shared nodes supporting continuity across different materials. Mixed dielectrics (e.g. Si, SiO_2, Si_xNi_y) and various boundary conditions are supported. Coyote Systems' software supports arbitrary layers and non-manhattan geometries making it suitable for most MEMS devices.

The mesh is iteratively refined with p- and h-adaptive meshing to accurately solve the field equation to a user specified accuracy. Areas where either the state or gradient of the solution vary greatly are refined, using either constant, bilinear, biquadratic or bicubic elements.

The electrostatic field is first solved calculating the unknown potentials or fluxes. The solved fluxes allow high-accuracy capacitance coupling and electrostatic force calculations. The electrostatic fluxes are "inherited" as a mechanical pressure which is applied to the mechanical model. The mechanical model, with its own optimal mesh, is solved to obtain the displacement and tractions. The electrostatic model "inherits" the mechanical displacement and the cycle iterates until convergence. Convergence is typically specified as success conditions (e.g. small change in capacitance, small change in displacement) and failure conditions (e.g. collision, maxtime, maxiterations).

The simulation of the TI micromirror is shown with all these steps in Figure 2. A converged simulation was automatically generated and solved in 20 minutes on a PC. As stated earlier, the Analog Devices accelerometer ADXL76 has been simulated in less than 1 minute on a PC.

1.4 Summary

The technology designed, developed and delivered shows that the approach chosen by Coyote Systems had technical merit and resulted in dramatic improvements in MEMS analysis capabilities of large, realistic MEMS devices. All technical issues encountered have been addressed and were either solved or circumvented. Coyote has met all milestones by the end of the program.

Coyote emphasizes that all work agreed to in the Agreement was fulfilled to the agreed budget cost, and that Coyote met all contract commitments.

All developed technologies have been commercialized with Coyote's AutoMEMS software for solving large, realistic MEMS devices.

2.0 Introduction to MEMS Simulation

This section describes the direction that Coyote Systems chose to enable the simulation of large, realistic MEMS devices. Coyote recognizes that many users are not simulation experts, and therefore Coyote provides robust accurate automated tools. The advantages provided by these tools are discussed in this section.

Of considerable commercial interest is the fact that the automated adaptive PDE solvers generated in this contract can also be used for variety of other problems (e.g. integrated circuit extraction, rational drug design).

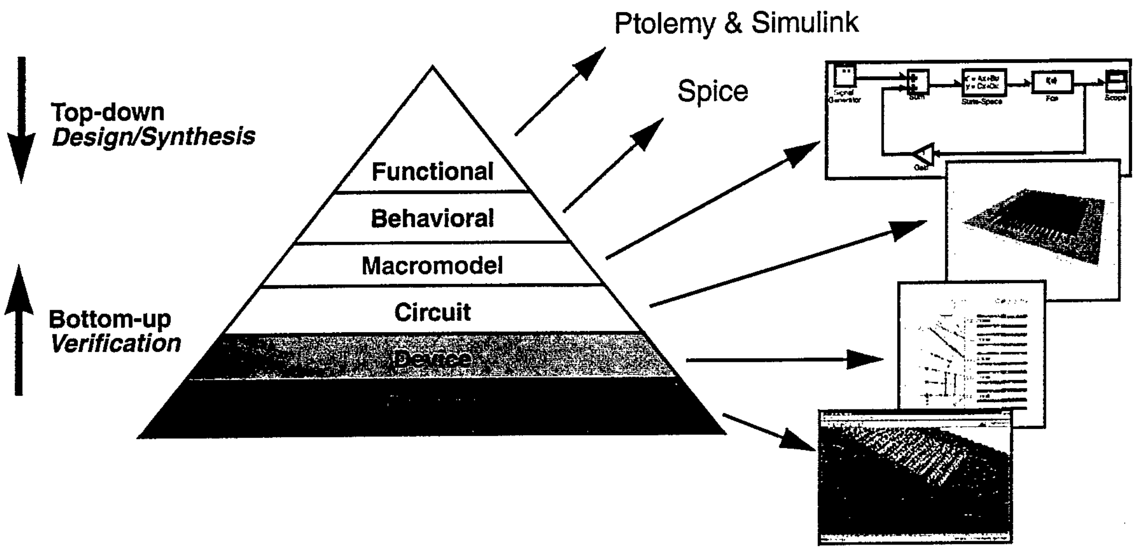
2.1 MEMS Simulation

This program developed software to effectively simulate MEMS devices by solving coupled 3D electrostatic and elastostatic fields using the boundary element method (BEM). This approach emphasized accurate and computationally efficient physics-based PDE analysis of large, realistic MEMS devices. These algorithms allow for robust automated analysis, eliminating the need for a "simulation expert".

Capabilities Coyote has successfully demonstrated include:

- automatic 3D model generation from 2D mask
- specification of boundary conditions using mouse to "point&click" on model
- automatic BEM discretization of model
- automatic BEM solution including p- and h-adaptive mesh refinement using error indicators
- automatic creation of table-based macromodels from BEM simulations
- high-level Simulink simulation of MEMS devices and closed-loop feedback

FIGURE 3. Bottom-up hierarchical MEMS analyses demonstrated by Coyote



As shown in Figure 3, hierarchical capabilities include:

- Physics=3D PDE analysis (e.g. electrostatics, elastostatics, thermal, thermoelastics)
- Device=analysis of MEMS component (e.g. comb-finger assembly)
- Circuit=coupled multi-physics analysis of entire MEMS device (e.g. accelerometer)
- Macromodel=reduced order multi-physics model (e.g. accelerometer Simulink model)
- Behavioral=system simulation (e.g. accelerometer table-based model in Spice)
- Functional=closed-loop system simulation (e.g. Simulink feedback accelerometer system)

2.2 Numerical PDE solvers

Accurately simulating the behavior and fabrication of MEMS structures involves creating of a geometric model and solving the appropriate partial differential equations (PDEs). Typically, this requires numerically solving the 3D Laplace field equation for electrostatics and the 3D Navier equation for elastostatics. These equations are coupled since the electrostatic forces affect the elastostatic deformation. The resulting deformation again changes the electrostatic fields and forces. As a result, simulating MEMS devices entails solving coupled nonlinear PDEs.

Three methods can be used to numerically solve these PDEs:

- FDM (finite difference method)
- FEM (finite element method)

- BEM (boundary element method)

FDM allows for the simple implementation of complex models, but requires a very large number of nodes for accurate results of 2D and 3D simulations. It is also difficult to resolve complex geometric features on FDM grids. Also, FDM is not well suited for accurate computation of secondary fields such as electrostatic field distribution.

FEM and BEM are more useful for simulation of complex geometries and are discussed in more detail below.

2.2.1 FEM

FEM[1] is considered to be the traditional approach for solving PDEs. FEM can handle linear, nonlinear, time varying and coupled problems. The numerical procedure to solve PDEs using FEM is to: discretize the model, calculate matrix elements, assemble global matrices, apply boundary conditions, and solve the resulting linear system.

However, modeling using FEM can be difficult for the user, as FEM requires careful meshing of the simulation domain volume, a nontrivial task for complex 3D geometries. With FEM, a weak formulation of the PDE is numerically integrated together with the element polynomial shape function, resulting in a local influence coefficient. This procedure is repeated for all adjacent elements forming a sparse linear systems.

Programming a FEM tool is not a complex task for the vendor, however, solving the PDE from the user standpoint can be very complex and time consuming.

2.2.2 BEM

BEM[2] is also an established numerical method developed at the same time as FEM. BEM also handles linear, nonlinear, time varying and coupled problems. A common perception of BEM is that it cannot handle nonlinear problems, but with recent extensions[3], nonlinear problems are indeed solvable using only a surface discretization.

Similarly to FEM, the numerical procedure to solve PDEs using BEM is to: discretize the model, calculate matrix elements, assemble global matrices, apply boundary conditions, and solve the resulting linear system. As BEM requires modeling of only surfaces, a 3D model is much simpler for the user to create. Further, BEM solves for both the state and gradient using integral methods to yield very accurate results.

Because of advanced mathematics used, programming a BEM tool is a quite complex task for the vendor, however, solving the PDE from the user standpoint is very simple and fast. As a result, using BEM is simpler for the user, and the simulation results will typically be more accurate than other numerical approaches.

BEM as used by Coyote is described in detail in Section 5.1, "Direct BEM," on page 29 and Section 5.5, "Multipole Accelerated BEM," on page 32.

1. Turner, "Stiffness and deflection analysis of complex structures", J. Aero. Sci. v. 23, 1956
2. Jaswon, "Integral equation methods in potential theory - I", Proc. Roy. Soc. Lond, v. A275, 1963
3. P. Partridge, The Dual Reciprocity Boundary Element Method, Elsevier, 1992

2.3 Computational Scaling

Aside from the benefits of a single mesh and a simpler model, there are other advantages of using BEM. One such advantage is the computational scaling of a BEM simulation. Since both FEM and BEM can solve similar PDEs, it is attractive to select the method with the lowest user and computational costs.

For example, it is desired to solve a PDE inside a cube with both FEM and BEM to obtain comparable results. The FEM model is volumetrically discretized with n nodes per side, resulting in a total of n^3 nodes. A BEM model consists of surfaces, or approximately n^2 nodes. Table 1 shows the number of nodes, memory and cputime requirements for each approach. Direct, Iterative and Multipole indicate the type of linear solver used. As can be seen, the multipole BEM method results in significantly smaller models and lower computational resources.

TABLE 1. Comparison of Computational Scaling BEM vs. FEM

	BEM			**FEM**	
	Direct	**Iterative**	**Multipole**	**Direct**	**Iterative**
# Nodes	n^2			n^3	
Memory	n^4		$(n\log n)^2$	n^6	$n^{4.5}$
CPU Time	n^6	n^4	$(n\log n)^2$	n^9	$n^{4.5}$

To illustrate the differences between the two methods, consider a model solved with similar spatial resolution. For example, if the BEM model has $n^2 = 2500$ nodes then the FEM model has $n^3 = 125000$ nodes. The BEM model is quite small and can be solved in a few seconds using Coyote's Generation 3 software. The FEM model is quite large and may require an overnight run on a workstation.

The difference becomes even more pronounced for large problems. For example, if the BEM model has $n^2 = 200\times10^3$ nodes and the FEM model has $n^3 = 100\times10^6$ nodes. Note that such a FEM model is unrealistic even on today's supercomputers.

For typical MEMS models, such as the Analog Devices ADXL76 accelerometer, typical small BEM meshes results in $n^2 = 50\times10^3$ nodes while a comparable FEM model has $n^3 = 11\times10^6$ nodes.

Coyote has demonstrated solving very large models exceeding two million nodes, which corresponds to 3E9 FEM nodes. Clearly such large FEM models are completely impractical.

2.4 Coupled MEMS Analysis using Fast BEM

Most MEMS users using existing FEM software will spend 2-3 weeks manually creating a 3D meshed model, which is then simulated overnight. While multipole accelerated BEM can certainly reduce the simulation time, a primary goal of this effort is to reduce the manual modeling time by using surface-based modeling tools and surface-based 3D numerical field solvers.

By creating a fast numerical field solver for electrostatic and mechanical fields, Coyote could create a framework specifically tailored for coupled MEMS analysis. The final software AutoMEMS incorporates additional improvements to make the tool intuitive and useful for MEMS design by both beginners and advanced users.

A robust and easy method to generate 3D meshed models directly from layout is provided using:

- 3D model building using process description
- easy-to-use graphical process description

Extremely fast and accurate solutions to the coupled PDEs are provided using the following robustly automated technologies:

- BEM multi-physics PDE solver
- adaptive BEM element refinement and error indicators
- acceleration using multipole BEM
- acceleration using constrained BEM
- acceleration using “tunnelling”

The tools are made easy-to-use with the most advanced GUI concepts, including:

- "wizards" to assist users in creating coupling scenarios
- "placeholders" to assist users in selecting valid simulation entries

Each of these technologies is further described in Section 3.0, “Visualization,” on page 14, Section 4.0, “Model Generation,” on page 18, Section 5.0, “BEM Engine,” on page 29, and Section 6.0, “Software Releases,” on page 53. In combination, each of these technologies allows both beginner and advanced MEMS users to simulate large, complex MEMS designs in a time-efficient manner. This allows the users to concentrate on designing their specialized MEMS device.

2.5 Overview of Milestones

Milestones were identified which would capture the spirit of the fast, robust, automated MEMS analysis envisioned by Coyote.

The nine milestones in this project were successfully completed. Details of the technical work behind each milestone are summarized here, and additional information is available in Section 3.0 on page 14, Section 4.0 on page 18, Section 5.0 on page 29, and Section 6.0 on page 53.

2.5.1 3D Electrostatic

The high-speed high-accuracy simulation of 3D electrostatic fields is an essential part of a coupled MEMS simulator. Using the boundary element method (BEM), the BEM Engine numerically solves the 3D Laplace partial differential equation with arbitrary material properties, arbitrary geometries, and arbitrary boundary conditions.

2.5.2 3D Elastostatic

The simulation of the 3D elastostatic fields is another significant part of a coupled MEMS simulator. Using the boundary element method (BEM), the BEM Engine numerically solves the 3D Navier partial differential equation with arbitrary material properties, arbitrary geometries, and arbitrary boundary conditions.

2.5.3 Adaptive Electrostatic

To obtain an accurate solution to a PDE, it is essential that the model discretization is sufficient to capture the simulated effects. Both h and p mesh refinement is offered for electrostatics models. H-refinement subdivides the selected elements into smaller sized elements, and p-refinement increases the shape function order (e.g. constant, linear, quadratic, cubic) of the selected elements.

Coyote also developed an automatic adaptive meshing solution removing the need for an expert user. After an initial solution is obtained on an initial mesh, the software evaluates the error on every element, and then automatically refines the elements with the largest errors.

2.5.4 Fast Electrostatic

To solve the electrostatic fields on large realistic MEMS devices, it is essential that the computational efficiency of the simulator is very high. Therefore Coyote developed the fast multipole accelerated BEM Engine, resulting in a near-optimal O(NlogN) scaling with memory and cputime. This is considerably improved over the naive BEM which has a computational scaling of $O(N^2)$.

2.5.5 2D Mask → 3D Model

To provide an easy to use MEMS analysis system, it must also be easy to generate the 3D models. Coyote created a model generator that accepted industry standard layouts (e.g. CIF, GDSII, annotated GDSII) and a Coyote-proprietary process description to emulate MEMS fabrication steps resulting in a 3D surface-based model of the MEMS device. The surfaces of the model are discretized forming the initial BEM model ready for analysis.

2.5.6 Adaptive Elastostatic

To obtain an accurate solution to a PDE, it is essential that the model discretization is sufficient to capture the simulated effects. Both h and p mesh refinement is offered for elastostatic models. H-refinement subdivides the selected elements into smaller sized elements, and p-refinement increases the shape function order (e.g. constant, linear, quadratic, cubic) of the selected elements.

2.5.7 Fast Elastostatic

To solve the elastostatic fields on large realistic MEMS devices, it is essential that the computational efficiency of the simulator is very high. Coyote originally intended to develop a fast multipole accelerated BEM tool (similar to the electrostatics tool), but instead developed a constrained meshing algorithm that dramatically reduced the size of the elastostatic mesh. As a result, even though the computational efficiency of the elastostatic engine is O(N^2), the size of the model (N) is so small that the cost is acceptable.

2.5.8 CAD Model Import/Export

Coyote software includes a 3D model generator that creates multi-layer multi-material 3D models using 2D photolithographic masks and a process description as described in Section 4.0, "Model Generation," on page 18. This model generator supports CMP-style processes common to many VLSI and MEMS designs.

For additional flexibility, Coyote wanted to support other CAD tools by allowing the Coyote software to import and export geometric models. The Coyote software can import geometry built using the commercial ACIS computational geometry engine (e.g. AutoCAD, SolidWorks). The spheres shown in Figure 8 on page 19 were created in AutoCad, meshed using ACIS development tool, imported and simulated using the Coyote field solver software.

2.5.9 Coupled MEMS

The coupled electrostatic/elastostatic simulation of large, realistic MEMS devices requires computationally efficient solvers, an efficient nonlinear iteration scheme, and an efficient convergence detection scheme. All these components were delivered.

2.6 Project Tasks

Information is provided on the technology and software developed during the contract. The descriptions illustrate the progress, the validity of the strategic and technical approaches, and the unique value-added capabilities that are being developed by Coyote Systems to create easy to use, fast and adaptive simulation tools for large, realistic MEMS devices.

The project consists of four primary tasks: Visualization, Model Generation, BEM Engine, and Releases. A short description of each of these follows.

2.6.1 Visualization

Advanced 3D rendering is provided using the visualization toolkit (vtk) which was developed as part of the National Supercomputing Initiative. vtk is a portable graphics package available on Unix and Windows platforms.

The graphic user interface (GUI) uses tcl/tk in conjunction with vtk to provide a simple "point&click" interface suitable for beginner users. The use of GUI "wizards" with placeholders and templates assists users to create complicated coupled or adaptive simulations.

2.6.2 Model Generation

The model generation consists of model building and CAD import functionality. Several MEMS and VLSI designers have successfully used this front-end to perform their first coupled 3D field simulations. They feel that it is an excellent system to provide complex field analysis, and that it is particularly well suited for users who are not "simulation experts".

2.6.3 BEM Engine

The BEM Engine is the Coyote developed computational engine for all 2D and 3D field solves as well as coupled MEMS simulations.

2.6.4 Releases

Coyote released interim software releases every 6 months to our Beta users. Software released includes both Redhat Linux v5.1+ and Solaris 2.5+ executables with an embedded licensing manager.

The Coyote software documentation includes:

- tutorial manual
- reference manual
- technical manual

The tutorials contain sample MEMS simulations progressing from the basic to advanced simulation sets. The tutorials lead the user step-by-step through several simulations of actual MEMS devices.

The reference manual contains installation and licensing information, descriptions and usage of every GUI menu command.

The technical manual contains a description of the Coyote application programming interface (API), Coyote file formats, client/server information, and verification examples.

All Coyote software documentation is available online at http://www.coyotesystems.com/pdf/autobem.tutorial.pdf, http://www.coyotesystems.com/pdf/autobem.reference.pdf and http://www.coyotesystems.com/pdf/autobem.technical.pdf.

Coyote has registered the name "AutoMEMS®" for the commercial release of the coupled MEMS simulation tool.

TABLE 2. Tasks and Subtasks

Task	Subtask
Visualization	3D Graphics
	GUI
	GUI Wizards
Model Generation	Layout Description
	Process Description
	Importing 3D CAD Models
	3D Geometry Engine
	3D Graphics
BEM Engine	Direct BEM

Task	Subtask
	Electrostatic BEM
	Elastostatic BEM
	Nonhomogeneous BEM
	Multipole Accelerated BEM
	Constrained BEM
	Tunnel Acceleration
	Adaptive Mesh Refinement
	Adaptive Meshing Error Indicators
	Volume-Based Forces
	Modal Analysis
	Coupled Electro/Elasto Simulations
	Relaxation Coupling
	Multi-level Newton Coupling
	Efficient Coupling
Software Releases	Development System
	Client/Server Architecture
	Command Language
	Quality Control & Testing
	Documentation
	License Manager & License Keys
	Porting to other Platforms
	Distribution
	Training
	Software Releases
	Electrostatic Verification
	Elastostatic Verification
	Coupled Electro-mechanical Verification

3.0 Visualization

Visualization of the MEMS devices is critical to allow the beginner user to properly create the model and interact with the simulation results. The AutoMEMS software (in Generation 1,2 and 3 releases) displays the 3D model using portable graphics and allows the user to interact with the model using an easy-to-use graphic user interface (GUI). No significant problems were encountered in the development of these capabilities.

Coyote developed the visualization system using two open-source libraries: 3D OpenGL graphics are displayed using the vtk graphics library and tcl/tk is used to create the menu systems. Both of these libraries worked extremely well.

3.1 3D Graphics

The visualization graphics are based on the portable open-source graphics library vtk[1]. vtk is a very powerful 3D graphics system built on OpenGL graphics. Any hardware that features OpenGL acceleration is automatically exploited by a vtk application, including the Coyote AutoMEMS software.

Vtk graphics are available for Sun, HP, Linux and Windows platforms.

3.2 GUI

The graphical user interface (GUI) is based on the portable tcl/tk environment. The GUI is extremely simple to use. There are 4 menu selections (File, Edit, View, Solve) and 4 icons (zoom, pick, pan, rotate). The GUI makes extensive use of context-sensitive popup windows, menus, "wizards" (templates and placeholders) and 3D graphics.

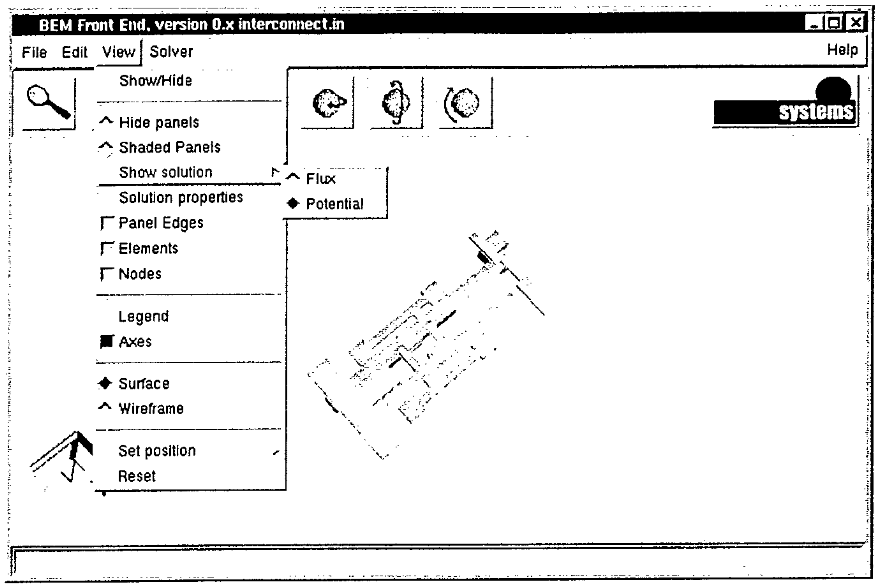
Using the GUI, the user

- specifies or creates a process description
- imports a geometry model,
- specifies material properties
- specifies boundary conditions
- solves the 3D fields
- visualizes and interrogates the results

In Figure 4 the user is displaying the color-coded 3D potential solution using the pull-down menu system.

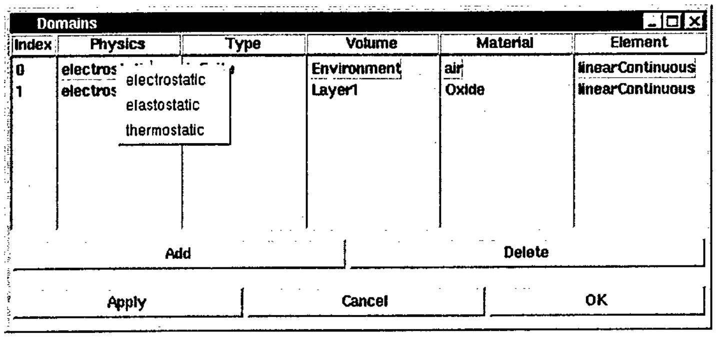
1. freeware distributed by http://www.kitware.com/vtk.html

FIGURE 4. GUI window to specify what solution component to visualize



In Figure 5, the user is specifying the type of physics (e.g. electrostatic, mechanical, thermal) for the simulation.

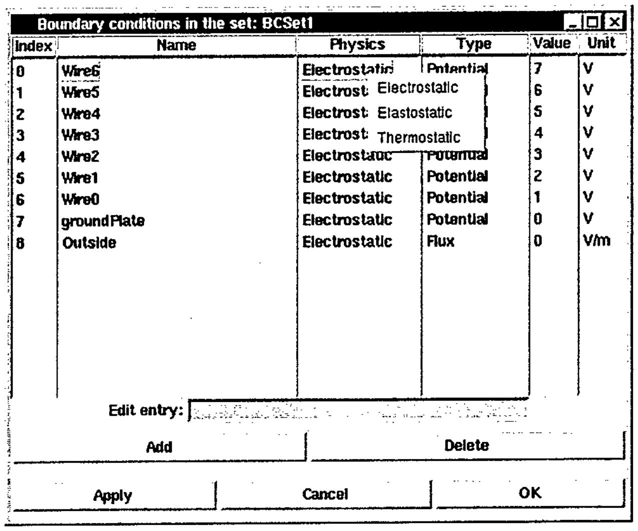
FIGURE 5. GUI window to specify domains of interest



Index	Physics	Type	Volume	Material	Element
0	electros		Environment	air	linearContinuous
1	electros		Layer1	Oxide	linearContinuous

In Figure 6, the user is specifying the boundary conditions on an electrostatic problem.

FIGURE 6. GUI window to specify boundary conditions



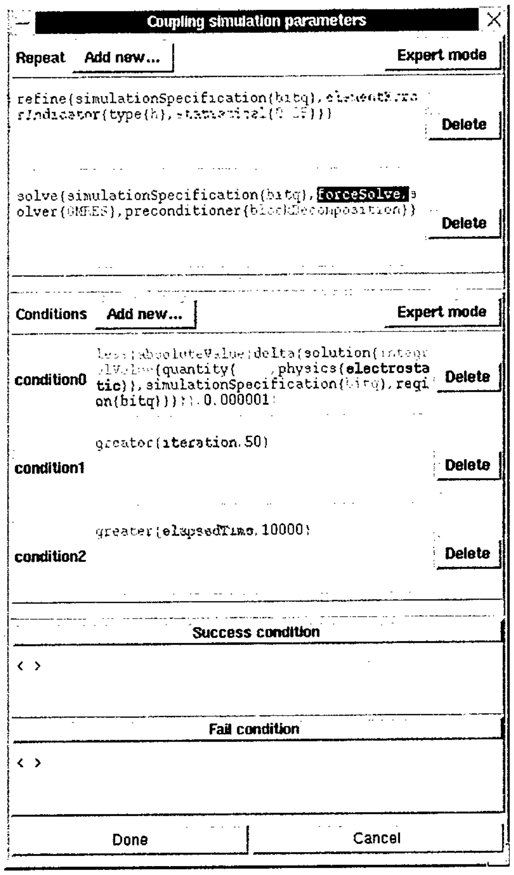
Index	Name	Physics	Type	Value	Unit
0	Wire6	Electrostatic	Potential	7	V
1	Wire5	Electrost		6	V
2	Wire4	Electrost		5	V
3	Wire3	Electrost		4	V
4	Wire2	Electrostatic	Potential	3	V
5	Wire1	Electrostatic	Potential	2	V
6	Wire0	Electrostatic	Potential	1	V
7	groundPlate	Electrostatic	Potential	0	V
8	Outside	Electrostatic	Flux	0	V/m

3.3 GUI Wizards

The GUI uses "wizard" templates and "placeholder" menu selections to simplify data entry. The wizard templates allow iterative adaptive refinement or iterative coupled simulations to be specified, presenting typical sequences of API commands to be performed. In each API command sequence, the user can choose from valid values encoded in color-coded placeholders. By clicking on a placeholder, a popup menu appears with all valid entries, allowing the user to choose one entry.

The combination of wizards and placeholders enables beginner users to quickly and accurately specify complex iterative commands.

FIGURE 7. Partially completed adaptive refinement wizard template. Selecting a color-coded placeholder reveals a popup menu with valid entries.



```
refine(simulationSpecification(bitq),elementErro
rIndicator(type(h),statistical(0.25)))

solve(simulationSpecification(bitq),forceSolve,s
olver(GMRES),preconditioner(blockDecomposition))

less(absoluteValue(delta(solution(integr
alValue(quantity(    ,physics(electrosta
tic)),simulationSpecification(bitq),regi
on(bitq))))),0.000001)

greater(iteration,50)

greater(elapsedTime,10000)

< >

< >
```

3.4 Summary

The GUI and wizards in AutoMEMS has shown itself to be easy to use for beginner and advanced users. The vtk-based graphics have shown themselves to be very powerful and extendable.

4.0 Model Generation

Given a 3D model, Coyote planned to develop the tools needed to mesh and solve the coupled fields on the model. Coyote had not planned for a significant model generation effort since other teams were developing such capabilities. Problems importing CAD models are described in Section 4.3 on page 26.

It quickly became apparent that the initial model generation and meshing were intricately linked, and that the same vendor should be providing solutions to both. Initially Coyote attempted to use a commercial geometry engine to create 3D models and mesh the resulting surfaces. Problems with computational throughput and software bugs in such systems made such a solution unacceptable. Problems using commercial geometry engines are described in Section 4.4 on page 27.

As a result, Coyote started a model generation effort which was included in the Generation 3 release. The Coyote model generator creates a 3D meshed model directly from a set of 2D photolithographic masks and a process description. By emulating a CMP-style fabrication process complex 3D models can be quickly generated. This solution is extremely fast and robust and supports most MEMS models. It would be desirable if future work augmented the Coyote model generator to also support non-planar structures.

4.1 Importing 3D CAD Models

Coyote has demonstrated the capability to import 2D and 3D ACIS *.SAT mechanical CAD from a variety of solid modeling tools (e.g. AutoCad v14, Bentley Microstation, SolidWorks).

The geometric entities in the SAT file cannot be used directly. Instead the surfaces of the SAT model are faceted using the ACIS solid geometry engine. Coyote has created scripts which automatically accomplish this meshing.

An arbitrary ACIS geometry model can be meshed using an economic third party ACIS application[2] (<$100). The resulting meshed file can be directly imported by Coyote. The application 3DScheme is an interpreted lisp-like language that includes most ACIS commands. By generating a scheme program, it is possible to read an existing ACIS geometry file, mesh it using ACIS commands, and save the meshed geometry in an output file.

As an example, the two spheres shown in Figure 10 were created using AutoCad, saved in ACIS format, meshed using 3DScheme, and imported by the Coyote software. The AutoCad geometry is saved in ACIS format using the AutoCad commands shown in Figure 8. The ACIS commands used by 3D Scheme to mesh the surfaces of the spheres are shown in Figure 9.

2. 3DScheme Pro v2 from Schemers Inc
http://www.schemers.com/3dspro20.html

FIGURE 8. AutoCad command to save geometry in ACIS format

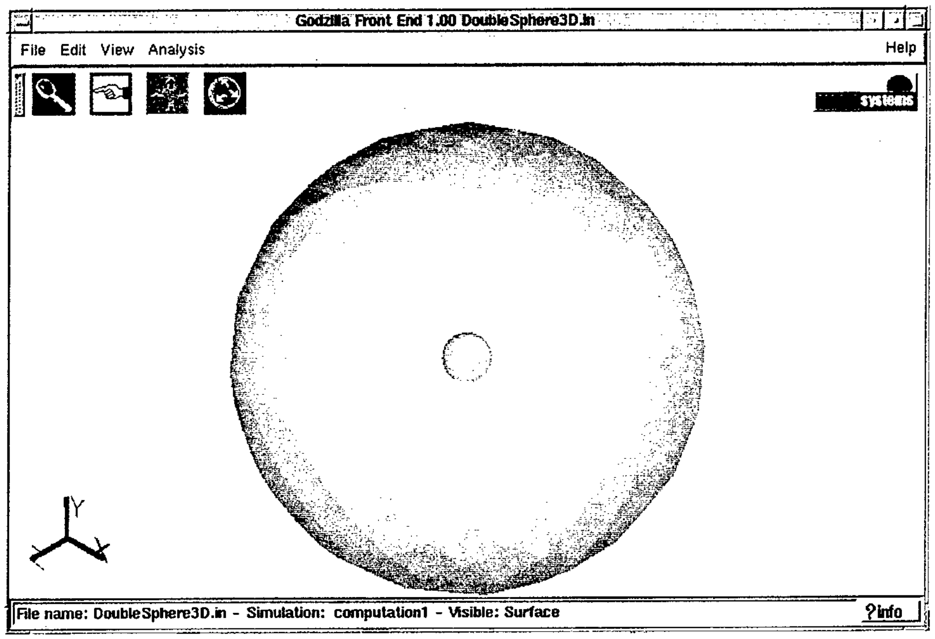
```
acisout file1.acis
```

FIGURE 9. 3DScheme script to mesh ACIS geometry file

```
(load "file1.acis")
(entity:facet (part:entities))
(entity:write-facets (part:entities) "file1.acis.mesh")
```

The resulting meshed geometry file can be directly imported by the Coyote AutoBEM software by specifying "*.*" as a filter in the File/Import menu. A sample meshed ACIS model is displayed in the Coyote GUI as seen in Figure 10.

FIGURE 10. Coyote model translated from AutoCad-to-ACIS-to-AutoBEM



File name: DoubleSphere3D.in - Simulation: computation1 - Visible: Surface

4.2 Model Generation by Fabrication Process Emulation

Coyote can create a 3D geometry from a 2D layout and process description file and create a Coyote-formatted inputdeck. By emulating the fabrication process, a 3D model can be automatically generated from the 2D layout masks. The 3D model generator has the following features:

- Emulates basic MEMS fabrication (uniform deposition, anisotropic etching) using a Coyote proprietary geometric engine
- Supports non-manhattan geometry
- Requires no user-interaction

4.2.1 Layout Description

The Coyote model generator accepts 2D layouts in GDSII, annotated GDSII or CIF formats. Any text labels in GDSII or CIF are automatically converted into region names enabling simpler geometric tagging and boundary condition specification.

4.2.2 Process Description

The process description is displayed and modified using a graphical representation shown in Figure 11. By clicking on any geometric entity, the geometric or material parameters can be edited, as well as adding new geometric entities.

For example, clicking on the bottom metal layer in Figure 11 results in context-sensitive popup window containing valid operations. Select "add dielectric" to add a dielectric as shown in Figure 12.

FIGURE 11. Process description window

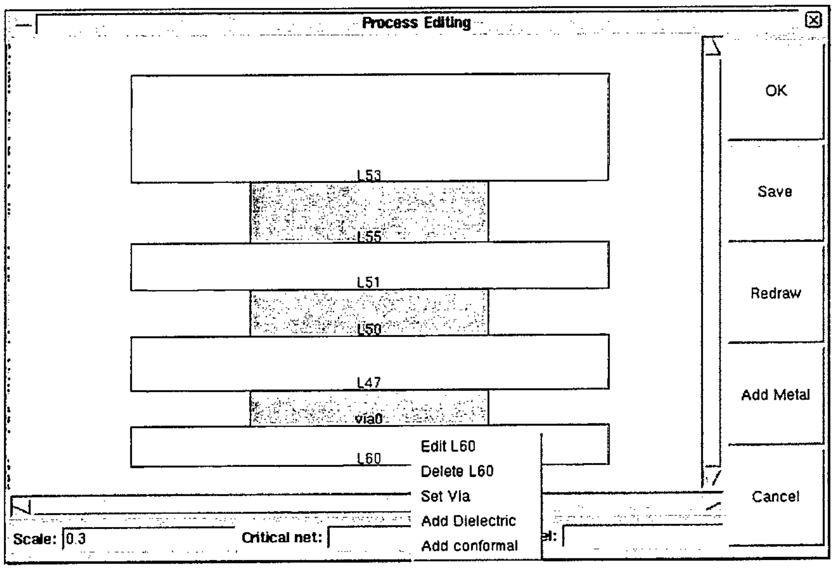
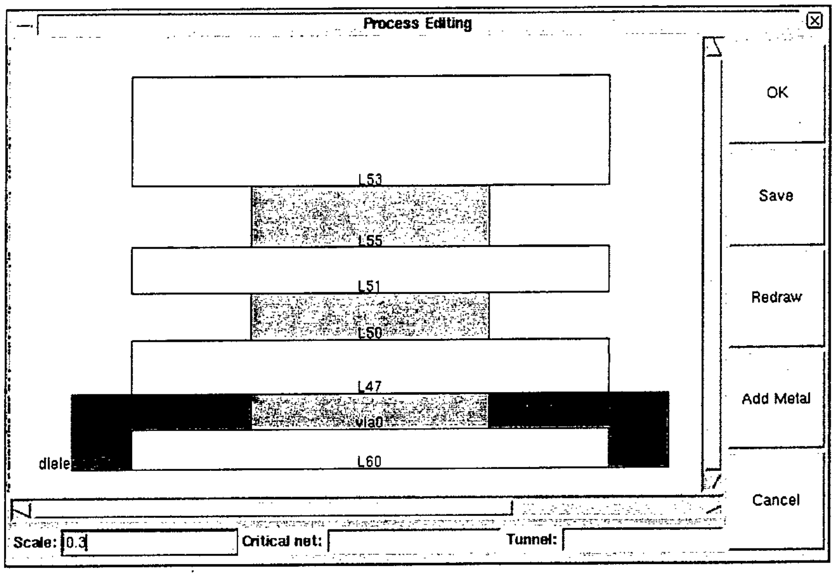


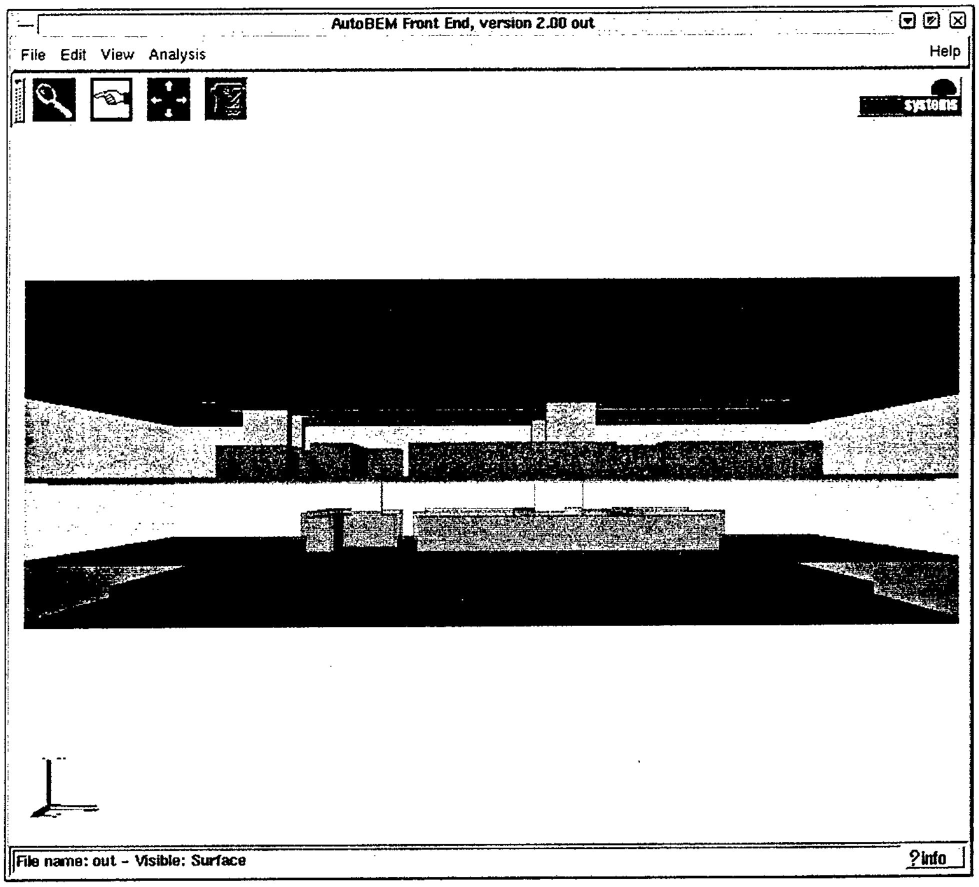
FIGURE 12. Added interlayer dielectric



Scale: 0.3 Critical net: Tunnel:

Similarly, conformal dielectrics can be added, and the resulting 3D model (for a SRAM layout) is shown in Figure 13.

FIGURE 13. 3D model with interlayer dielectrics and conformal dielectrics


AutoBEM Front End, version 2.00 out
File Edit View Analysis
Help
systems
File name: out - Visible: Surface
?Info

The graphical process description is automatically saved in a text file format, containing the bottom height, the thickness, and the material property of each layer, a generic planar geometry can be created. These parameters are defined in the process description in Table 3.

TABLE 3. Coyote process description

CIF	GDSII	Annotated GDSII	Process Description Command	Description
x	x	x	# <string>	comment
x	x	x	scale <value>	drawing scale in the graphical editor; not used in generation
x	x	x	metal <name> <height> <thickness> <resistivity>	
x	x	x	via <name of metal below> <name> <height> <thickness> <resistivity>	
x	x	x	dielectric <name of metal> <name> <height> <thickness> <permittivity>	
x	x	x	conformal <name of metal> <name> <height> <thickness> <permittivity>	
		x	tunnel <radius>	discards geometry outside tunnel
		x	criticalNet <name>	generate model and BC for aggressor net

An example of what a Coyote process description looks like is provided by the textual process description in Table 4 that used to create the 3D model in Figure 13.

TABLE 4. Process description of SRAM.PRO

```
scale 0.3
metal L60 -200 110 1.0
via L60 via0 -90 100 1.0
dielectric L60 oxide1 -200 210 4
conformal L60 conf1 -200 20 3.0
metal L47 10 150 1.0
via L47 L50 160 130 1.0
dielectric L47 oxide2 10 280 3
conformal L47 conf2 10 20 1.0
metal L51 290 130 1.0
via L51 L55 420 170 1.0
dielectric L51 oxide3 290 300 4
conformal L51 conf3 290 20 1.0
metal L53 590 300 1.0
dielectric L53 oxide4 590 400 1
conformal L53 conf4 590 20 1.0
```

4.2.3 3D Geometry Engine

Coyote originally used the commercial geometry engine ACIS, but due to bugs dealing with self-intersecting surfaces, Coyote was forced to create a simpler internal engine. Figure 14 shows a 2D mask with many overlapping tiles.

Figure 14 shows a 2D layout consisting of several overlapping manhattan tiles. The overlapping tiles are eliminated using 2D boolean operations. The resulting polygonal outlines are then extruded to form a 3D model shown in Figure 15.

The sides of the 3D model consist of quadrilateral panels, and the top polygonal surface is discretized using delaunay triangulation. An example of such a generated mesh is shown for the XL50 in Figure 16.

FIGURE 14. Section of a typical 2D layout mask showing overlapping quadrilateral boxes

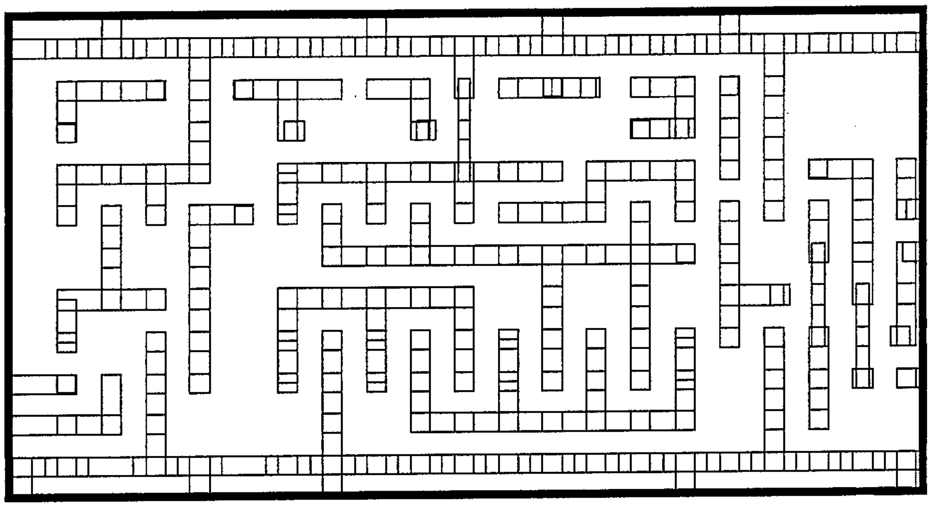
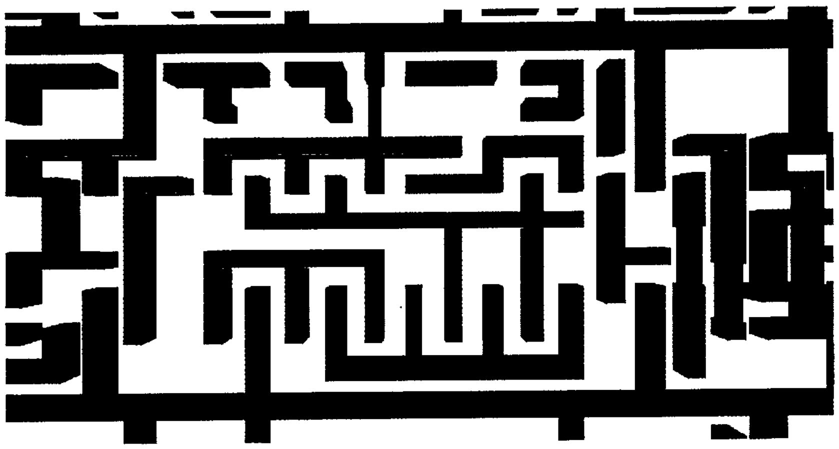


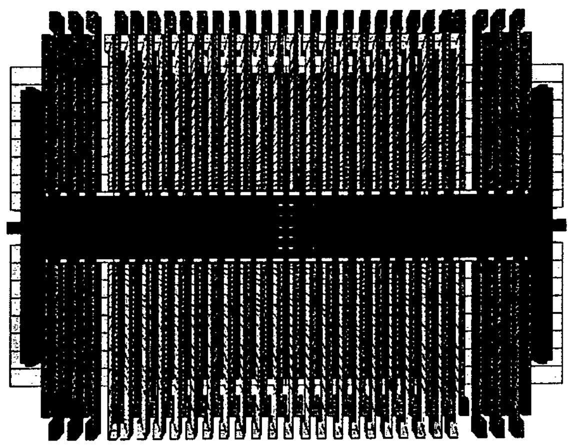
FIGURE 15. A 3D model is formed by extruding the 2D layout. The boxes were merged using boolean union operations.



By performing a boolean union on the overlapping tiles, a polygonal boundary representation is obtained. These polygons are then extruded to form a 3D geometry as shown in Figure 15. The extruded sides are discretized into quadrilateral BEM panels, and the top and bottom surfaces are discretized into triangles using Delaunay meshing. This is a high capacity solution, capable of processing 1 MB of GDSII data in 1 minute on a PC.

This capability was used to create a 3D model and mesh of the ADXL76 accelerometer shown in Figure 16 in 10 seconds of cputime. The mesh was refined as shown in Figure 28.

FIGURE 16. The 3D model of ADXL76 formed from the masks is meshed for CBEM.



4.3 Problems with CAD Import

Many common CAD packages (e.g. AutoCAD, Microstation) can export their geometries in SAT format, and we believe that supporting the ACIS SAT format would be a useful functionality to support arbitrary geometric models.

Common mechanical engineering CAD geometry formats include DXF and IGES. Coyote has written and tested import routines that support a subset of these formats. DXF 2D/3D polylines and 3D faces are supported. IGES 2D/3D lines and 3D FEM elements are supported. Note that there are literally hundreds of different DXF and IGES geometry types, and it is not feasible to support them all. For example, if a drawing in Autocad is saved in IGES format, Autocad cannot read the resulting IGES input file. Similarly, there are dozens of FEM elements defined, but no BEM elements. Because Coyote is working with BEM, this limits IGES compatibility.

It would be desirable to import any geometry (e.g. DXF, IGES, SAT) and automatically BEM mesh and analyze the device. Because of the various DXF and IGES formats, Coyote will not attempt this. Instead, Coyote will offer full support of the ACIS SAT geometry format.

In 99Q3, ACIS representatives admitted that the geometrical complexity of the VLSI and MEMS models was beyond their current capabilities, and would be for the foreseeable future. ACIS is capable of creating toy MEMS models, but the computational efficiency is poor prohibiting the use of ACIS to create models that include multiple layers of non-planar structures and etch-holes. In these cases the ACIS engine typically calculates for several hours only to result in segment overflows and crash. Further, Coyote identified bugs when ACIS generated illegal self-intersecting models, and these bugs have still not been fixed by the commercial vendor. Coyote is not providing any built-in support for ACIS models, but will help customers create or import models on a consultancy basis.

Because of the high cost of an ACIS development license (>$60k) and poor runtime performance of ACIS, Coyote has not obtained a license. Instead, Coyote has used a free evaluation license. Coyote has identified a low-cost interpreted language application that features an ACIS implementation that can be used to mesh user-created ACIS models.

4.4 Problems with process emulators

A versatile method of generating an initial 3D model of the MEMS device from a 2D mask layout is needed. Rather than simply extrude the mask shapes into 3D polygons, a more realistic method which emulates or simulates the process fabrication flow was desired. Several such emulators and simulators were identified. One of the most promising emulators was 3DEYE[3] from University of Edinburgh which generated BEM models for IC extractions. Unfortunately, after evaluation this software could not be extended to support non-manhattan geometries and realistic sidewall shapes. Further, the cputime was excessive to model even small geometries.

4.4.1 Commercial geometry engine

Coyote Systems identified that a general geometry engine would be needed to generate 3D boundary representations using boolean operations. The commercial geometry engine ACIS[4] was investigated to emulate MEMS material deposition and etching. We desired to generate conformal depositions (i.e. added material has constant thickness measured along the normal of the surface) and rigid depositions (i.e. added material has constant thickness measured along the normal of the wafer). We have been able to generate conformal depositions and etching simulating curved arbitrary shapes. However, even lengthy consultations with the ACIS technical support and consultants, we have not been able to generate rigid depositions even for manhattan geometries.

To emulate material deposition, we identify all visible surfaces of all materials and "grow" the surfaces along their normals. In some cases, these new surfaces may form self-intersecting geometries forming non-manifold solids. The ACIS tools do not currently identify and remove such non-manifold geometries. Any boolean operations on such non-manifold solids are not allowed by the ACIS toolkit limiting its functionality.

Because of the bugs in ACIS, Coyote did not acquire an ACIS license. Spatial Technology, the developer of ACIS, reported in 1998 that these bugs were significant and would not be fixed for quite some time. At the end of this program, Spatial has still not fixed these bugs, and Spatial has agreed that ACIS is not suitable for the large geometries that are common in MEMS or VLSI.

4.4.2 Coyote Solution

Coyote has successfully developed a simpler and more cost-effective method to generate a non-conformal 3D model from a 2D layout (instead of using ACIS). This process emulation method can be included in Coyote's portable C++ libraries without any fees,

3. http://www.ee.ed.ac.uk/profiles/research/STR/research_projects/SERC_3D/SERC_3D.html
4. Spatial Technology, Boulder, CO, http://www.spatial.com

and is suitable for extremely large data sets. This capability will be included in the next software release.

The only significant disadvantage is that the 3D models will only simulate CMP-style planar structures. Non-planar structures are not supported in the current software release, but could be added if the geometry engine was rewritten.

4.5 Summary

The automatic 3D meshed model generation starting from layout has shown itself to be very efficient and easy to use for large complicated MEMS devices. The only significant shortcoming is that each structural layer (e.g. polysilicon, metal) must be planar. Typical features such as "step-up anchors" are not realistically recreated.

5.0 BEM Engine

Coyote early decided that the largest cost involved with a coupled MEMS simulation was not the analysis engine but the model generation and meshing required to prepare for a simulation. Experience within Coyote had shown that an expert finite element method (FEM) user could create an accurate FEM model in about 2 weeks of engineering time, and that this model could be simulated overnight. Coyote worked to automate the model generation and meshing to dramatically reduce the total engineering time such a simulation would take. This was accomplished by using a boundary element method (BEM) implementation for both electrostatics and mechanical (elastostatic) 3D fields. As a result, only the surfaces of the 3D models need to be discretized which can be easily automated.

In addition, Coyote expended considerable effort to deliver a BEM analysis engine that could adaptively refine the mesh to deliver a user-specified accuracy. Doing so would remove the "numerical analysis expert" from the simulation flow and allow "normal" MEMS designers to simulate and optimize their designs. Further, Coyote worked to deliver a very high-speed electrostatic BEM engine since most of the MEMS simulations required accurate electrostatic force and cross-capacitance simulation results.

The Coyote design decisions are described below, starting from a direct BEM implementation to an accelerated BEM engine several orders of magnitude faster. Different schemes for coupled electro-mechanical simulation were also investigated, and the most computationally efficient method was chosen. Adaptive refinement and error indicators are also described which allow the user to specify a maximum relative error in the simulation results.

The delivered BEM engine works well for most MEMS designs. Extremely large electrostatic simulations are fully supported, and coupled simulations for typical MEMS problems (e.g. micromirrors) are straightforward to simulate. It would be interesting for future work to also accelerate the mechanical BEM solver to allow larger mechanical simulations and to add capabilities for modal analysis and residual stress analysis.

The computational engine that allows high-speed high-accuracy coupled simulations relies on an extremely efficient implementation of the boundary element method (BEM). A BEM solver is used for electrostatics and a second BEM solver is used for elastostatics. The electrostatic BEM solver is accelerated using multipole acceleration to provide greatly reduced computational scaling and reduction in cputime.

5.1 Direct BEM

Because the Boundary Element Method (BEM) only discretizes the boundaries or surfaces of the 2D or 3D geometry, the resulting BEM model is both very compact and easy for the user to generate. Practically any 3D solid model can be automatically robustly discretized into a valid BEM model.

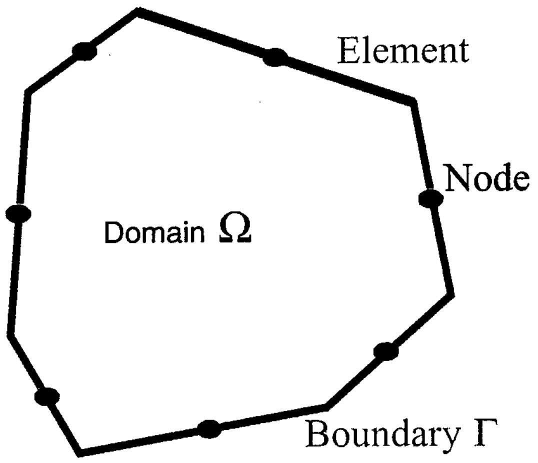
BEM has the capability to solve linear, non-linear and time-varying partial differential equations (PDE). BEM solves for both the state and gradient of a PDE field. For example, in a electrostatic problem, the state is the potential and the gradient is the electrostatic flux. In a thermal problem, the state is the temperature and the gradient is the heat flux. In a elastostatic problem, the state is the displacement (x, y, z components) and the gradient is the traction (similar to stress with x, y, z components). Thus BEM allows extremely accurate results when the user is interested in capacitance or stress concentration factors.

The Boundary Integral Formulation uses Green's Functions to describe the effects of loadings on the entire domain. Using the divergence theorem, the BIE is written in terms of integrals on the boundary of the domain

$$c \cdot u(\xi) = \int_{\Gamma} q(x) \cdot u^*(\xi, x) \cdot dx - \int_{\Gamma} u(x) \cdot q^*(\xi, x) \cdot dx \qquad \text{(EQ 1)}$$

with the known state boundary condition $u(x) = \bar{u}(x)$ on the Dirichlet boundary $x \in \Gamma_D$, and the known gradient boundary condition $q(x) = \bar{q}(x)$ on the Neumann boundary $x \in \Gamma_N$. A simple 2D BIE model is shown in Figure 17.

FIGURE 17. 2D BEM model with constant elements



The integral coefficients for each boundary segment are calculated for all node pairs. This step is very computationally expensive because the direct BEM method calculates interactions with all node pairs.

The coefficients in Eq. 1 can be assembled into a dense matrices describing the effect of the states and gradients.

$$H \cdot \begin{bmatrix} u_0 \\ \dots \\ u_{N-1} \end{bmatrix} = G \cdot \begin{bmatrix} q_0 \\ \dots \\ q_{N-1} \end{bmatrix} \qquad \text{(EQ 2)}$$

The known states and gradients are inserted into the above equation resulting in a standard linear system which can be solved.

$$A \cdot \begin{bmatrix} x_0 \\ \dots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ \dots \\ b_{N-1} \end{bmatrix} \qquad \text{(EQ 3)}$$

Given a BEM solution on the boundary of the domain, postprocessing can be used to calculate the state or gradient on arbitrary points in the domain or on the boundary. This capability is used to evaluate error-estimators on the boundary.

Note that the matrices in Eq. 4 are dense, which results in a O(N^2) scaling of cputime and memory. As a result, the direct BEM presented in Section 5.1 "Direct BEM" is not suitable for large models, instead some matrix sparsification such as that in Section 5.5 "Multipole Accelerated BEM" is required to solve large models.

5.2 Electrostatic BEM

Specialized Green's functions for 2D and 3D Laplace problems were used to solve for electrostatic fields. First a direct BEM approach was implemented and verified, after which specialized multipole acceleration operators were developed for all BEM integrals.

The capabilities of the solver were augmented to support arbitrary materials and arbitrary boundary conditions (Neumann, Dirichlet, mixed, floating potential). Extensive effort was spent on creating efficient algorithms and datastructures to support large models, and conversely, so quickly simulate small and medium sized models.

5.3 Elastostatic BEM

Specialized Green's functions for 2D and 3D Navier problems were used to solve for elastostatic fields. First a direct BEM approach was implemented and verified. Note that the elastostatic problem is a 3x3 tensor compared to the scalar electrostatic problem. A direct BEM elastostatic problem will therefore require at least 9x the cputime compared to a similarly sized electrostatic problem.

Initially specialized multipole acceleration operators were planned to be developed, but this proved to challenging. Instead Coyote realized that the mesh for the mechanical simulations could be substantially reduced as described in Section 5.6, "Constrained BEM," on page 36, allowing the O(N^2) computational scaling to be acceptable.

5.4 Nonhomogeneous BEM

Dual reciprocity boundary element method (DR-BEM) was used in Generation 1 release to demonstrate that non-linear and non-homogeneous partial differential equations could efficiently be solved using a BEM approach.

Several non-linear and non-homogenous PDEs were successfully simulated using DRBEM with Generation 1 release. However, the concentration of effort on future software releases was on fast analysis of large, complex, realistic MEMS models. As a result, most of our effort was directed towards efficient algorithms and datastructures. Because DR-BEM requires matrix operations on several matrices (including those in Eq. 4), this requires that the dense matrices be explicitly formed. Because of the size of realistic MEMS models, forming the dense matrices is impractical on any supercomputer. For example, the ADIXL76 accelerometer is typically meshed with about 35,000 panels for an electrostatic analysis. A DR-BEM analysis would require approximately 10 matrix operations on matrixes of size (35E3 x 35E3). Just storing the floating numbers (80 bits or 10 bytes) in these matrices would require 10 x 35E3 x 35E3 x 10 = 122GB, which is clearly impractical. As a result, this functionality was not included in the final software Generation 3 release.

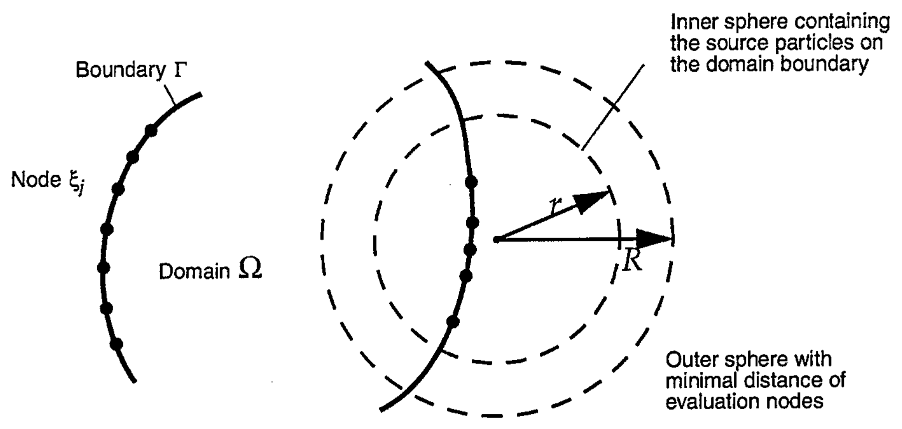
5.5 Multipole Accelerated BEM

To enable rapid solutions, Coyote uses the multipole accelerated fast BEM (FBEM) method which scales as $O(n\log n)$. This allows large, realistic geometric models to be rapidly simulated on standard PCs. To guarantee accurate solutions, Coyote uses error indicators and adaptive meshing to automatically refine the BEM model to a user specified solution accuracy on the state and gradients.

Compared to our initial work, Coyote has further reduced the memory and cputime used of FBEM by approximately 50% compared to BEM.

The direct BEM can be radically accelerated by using FBEM which clusters distant nodes together eliminating the need to calculate all the boundary integrals. Strict error bounds allow very accurate results. The boundary segments are represented by multipole expansions.

FIGURE 18. Clustering of distant nodes reduces required number of computations


Boundary Γ
Node ξ_j
Domain Ω
Inner sphere containing the source particles on the domain boundary
r
R
Outer sphere with minimal distance of evaluation nodes

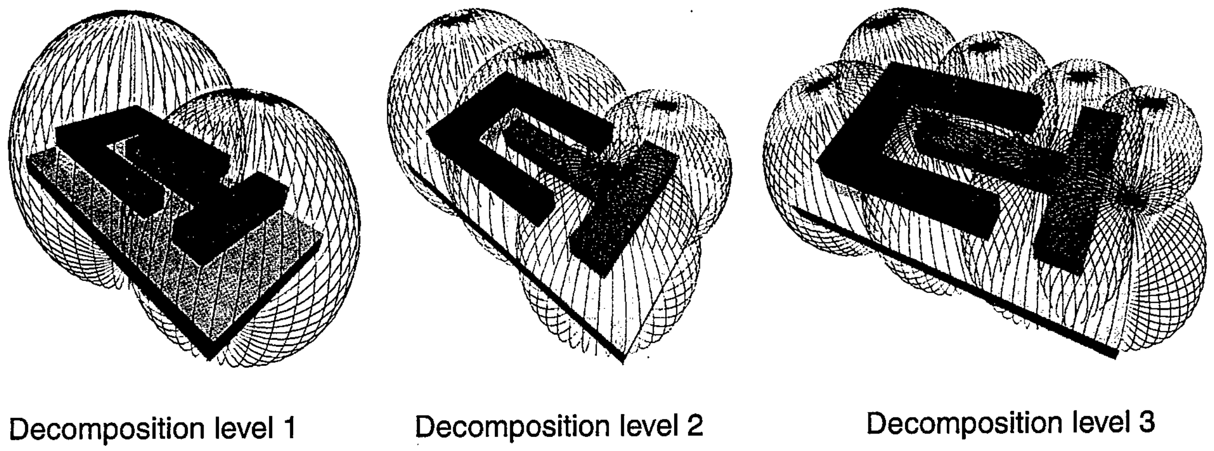
The distant clustered boundary integrals are now computed using MP expansions

$$\int_{\Gamma'} q(x)u^*(\xi, x)\,dx \cong \sum_{n=0}^{p} a_n \cdot M_n(\xi - c) \qquad \text{(EQ 4)}$$

This allows Integral evaluation by clusters instead of one element at a time. This introduces sparsity into BEM radically reducing the computational effort and cputime required for accurate analysis.

The domain is decomposed into hierarchical clusters, and MP clusters are formed for each cluster. The MP coefficients depend only on the MP coefficients of the lower cluster.

FIGURE 19. Hierarchical decomposition of nodes into clusters



The FBEM results in substantially faster computation (See Figure 20) and memory savings (See Figure 21). Note that instead of the BEM computational scaling of

$$O(N^2) \qquad \text{(EQ 5)}$$

the FBEM has the much better computational scaling of

$$O(N \log N) \qquad \text{(EQ 6)}$$

where N are the number of BEM nodes. From the graphs, it is apparent that large FBEM problems scale almost linearly. In other words, a model twice as big takes twice as long to solve.

FIGURE 20. Cputime for BEM coefficient calculations and equation assembly.

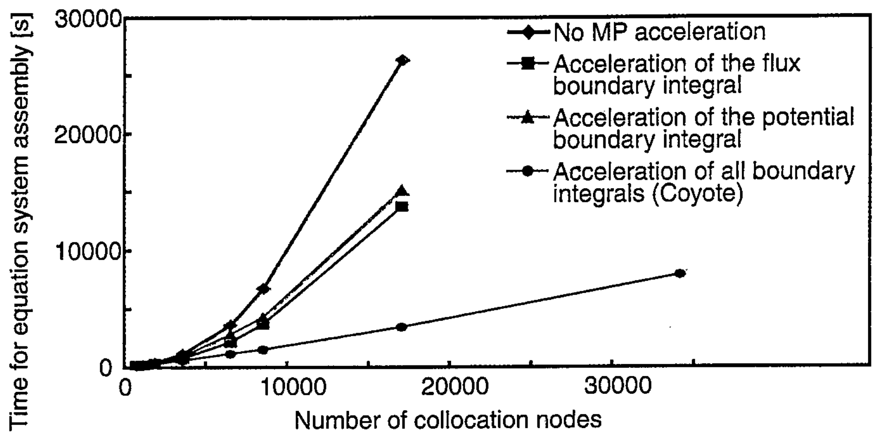
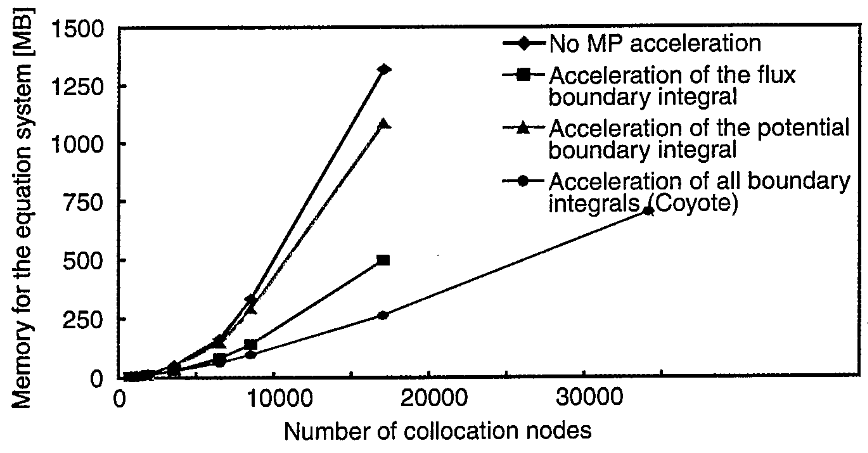
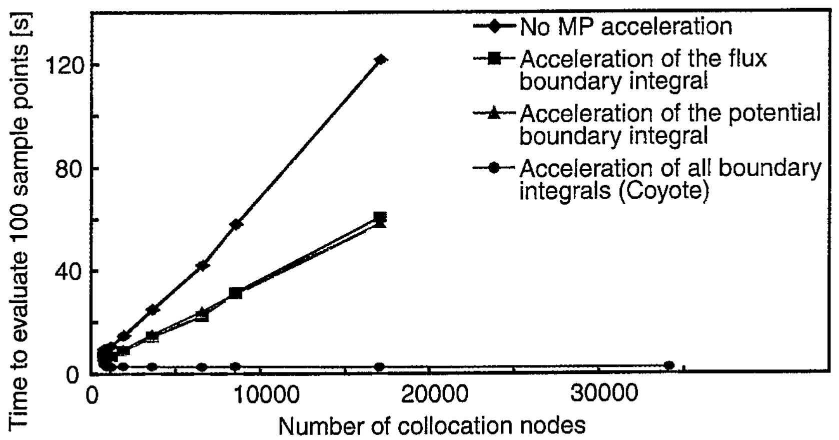


FIGURE 21. Required memory for direct BEM and fast multipole BEM



The cputime for postprocessing is also radically reduced as shown in Figure 22. This capability is used to evaluate error estimators on the boundary. Depending on the error estimator, the boundary elements are subdivided into smaller elements or higher order polynomial elements to ensure an accurate solution. Note that the sample point evaluation cost is independent of the simulation size.

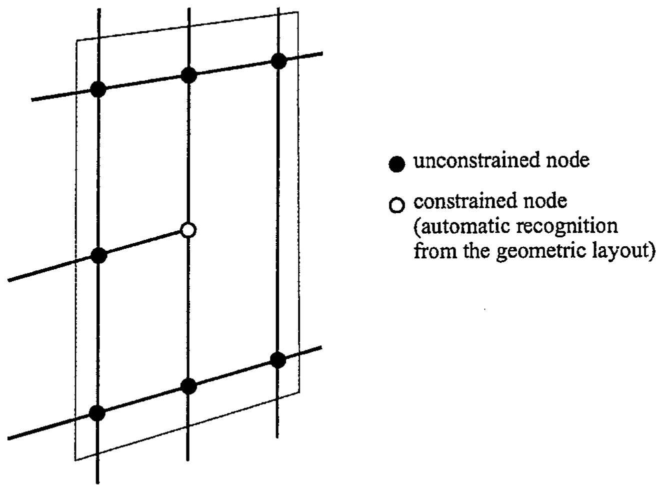
FIGURE 22. Cputime required to evaluate sample points



5.6 Constrained BEM

Constrained mesh refinement provides the benefits of both continuous and discontinuous BEM meshes. This allows for simple discontinuous subdivision of existing elements (See Figure 23) while still resulting in continuous nodal solutions. The Constrained Boundary Element Method (CBEM) identifies and solves for unconstrained nodes in a manner similar to BEM, while the identified constrained nodes are assigned nodal values based on interpolation of the solved constrained nodes. As a result, the degrees of freedom of the system are reduced to the number of unconstrained nodes, typically reducing the DOF by a factor of 4-6. This implies that the cputime and memory requirements are also reduced by 4-6x with no accuracy loss.

FIGURE 23. Coyote has developed a Constrained Boundary Element Method (CBEM) that allows discontinuous meshes by using constrained nodes. This enables very easy mesh refinement while still resulting in continuous states and gradients.


● unconstrained node
○ constrained node (automatic recognition from the geometric layout)

CBEM automatically identifies constrained and free nodes from a 3D mesh as shown in Figure 25.

FIGURE 24. 3D CBEM model showing how mixed order and mixed size elements can legally be used.

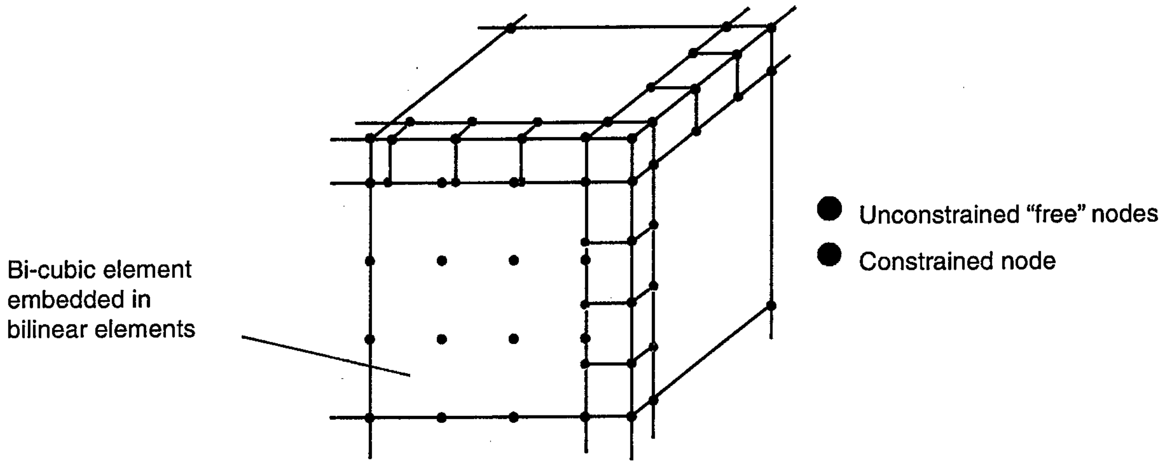
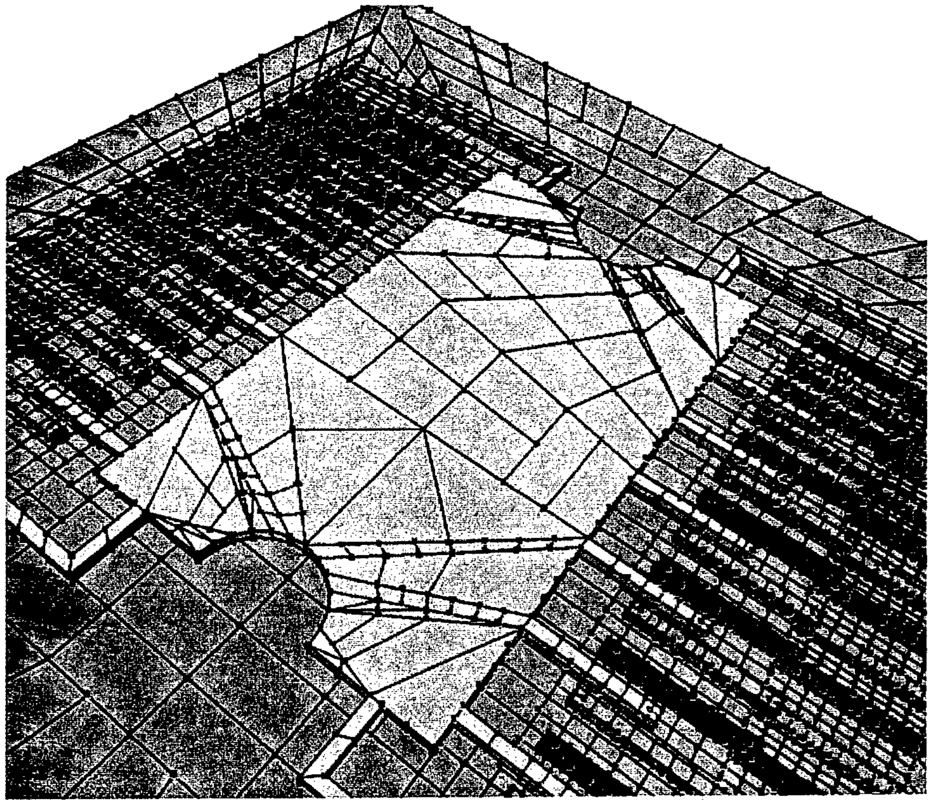

Bi-cubic element embedded in bilinear elements
● Unconstrained "free" nodes
● Constrained node

FIGURE 25. CBEM model of accelerometer reduces degrees of freedom by 6x, which reduces the cputime by 4x compared to fast multipole accelerated BEM.



● Unconstrained "free" nodes

● Constrained node

5.7 Tunnel Acceleration

While fast constrained BEM has a very good computational scaling of O(NlogN), the number of BEM panels (N) can still be very large for large, complex models. To achieve even faster computations, Coyote created a "tunnel acceleration" which automatically selectively eliminates geometry which does not significantly contribute to the solution.

BEM solves for a field and enforces that the field value is zero at infinity. For many problems, the field's region of interest is much closer to the geometry than infinity, or even 20 microns away. After heuristically determining a tunnel radius, the field solution on the tunnel surface should be zero (to approximate infinity). If the result is not negligible, then the tunnel was too small and the method is repeated.

Using the tunnel acceleration allows dramatic memory and cputime reductions of 40x for some problems, while others are not accelerated at all. Coyote has designed, implemented and verified the tunnel algorithms for electrostatics to result in less than 1% errors.

The electrostatic field was solved on the interdigitated comb-fingers allowing capacitance coupling and electrostatic force calculations. Various dielectric domains are supported, such as infinite surrounding space, silicon oxide and silicon nitride. Various applied boundary conditions are supported, including Neuman, Dirichlet, mixed and floating. The automatic model generation and analysis of this complex ADXL76 accelerometer was completed in 1 minute on a 400 MHz Pentium II PC.

FIGURE 26. Tunnel around magenta/yellow driving comb-fingers in ADXL76. The blue/purple sense comb-fingers are not included in the tunnel making the BEM model significantly smaller. This tunneled BEM problem is solved in 35 seconds.

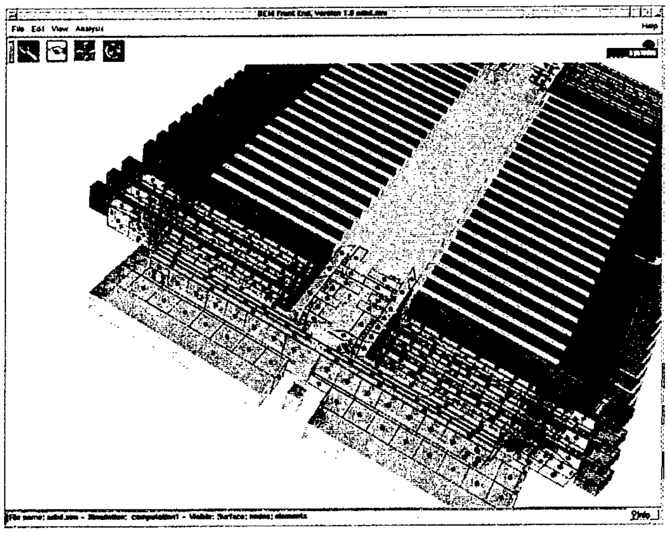
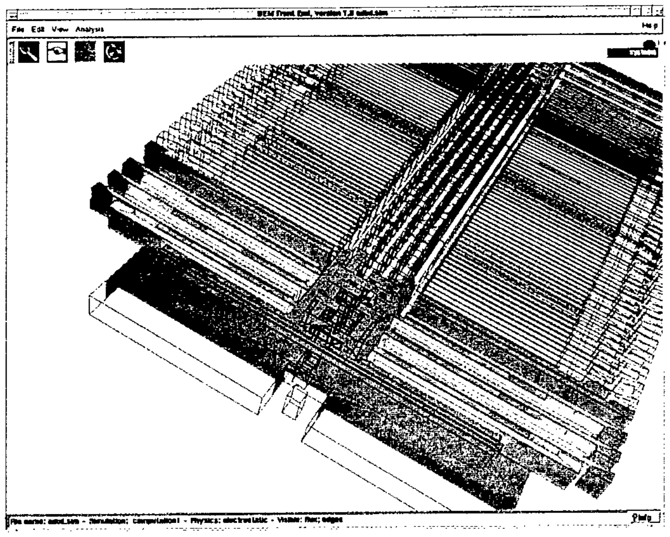


FIGURE 27. The calculated charge on the drive comb-fingers is displayed. Experiments have shown that typically <1% errors are incurred using tunnel acceleration.



5.8 Adaptive Mesh Refinement

After an initial solution is obtained, the user can adaptively refine the elements to obtain extremely accurate field solutions. By iterating with several refinements, the user can be assured of an accurate and fast PDE simulation.

The elements to be refined are identified in the refinement condition. These conditions can be:

- all
- element size
- aspect ratio
- error indicator
- maximum diagonal
- element smoothness
- element tunnel

5.8.1 Refine all elements

All elements are refined.

5.8.2 Element Size

All elements exceeding the specified maximum size are refined. In 3D, the size is equivalent to area, in 2D the size is equivalent to length. To refine 3D elements based on length, use the maximum diagonal refinement.

Adjacent elements with sizes varying more than an order of magnitude can introduce numerical problems.

5.8.3 Aspect Ratio

All elements exceeding the specified maximum aspect ratio are refined.

We have found that acceptable aspect ratios are less than 50:1 for medium accuracy solutions, and less than 20:1 for very high accuracy solutions.

5.8.4 Error Indicator

All elements exceeding the specified maximum error indicator are refined.

AutoBEM can compare the results of the BEM solution with additional integral evaluation points on every panel. By identifying the panels with the largest difference, this indicates which panels have the largest errors where refinement is needed. This error indicator technology is described in Section 5.9, "Adaptive Meshing Error Indicators," on page 42.

After identifying which elements should be refined, the user can resolve the new model. If the obtained solution is not sufficient, then this procedure can be iterated.

5.8.5 Maximum Diagonal

All elements exceeding the specified maximum element diagonal length are refined.

5.8.6 Element Smoothness

All elements exceeding the specified maximum element smoothness are refined.

"Smoothness" is a heuristic approach to refinement that will refine the areas around "sharp" corners. It is up to a user to define what will be considered "sharp" and what will be considered "smooth".

For a corner of a cube, a reasonable parameter can be any number less than 3 (because there are three sides meeting at the corner). Obviously, sharp corners cannot be smoothened in finite number of steps, hence some reasonable limit should be kept for the depth.

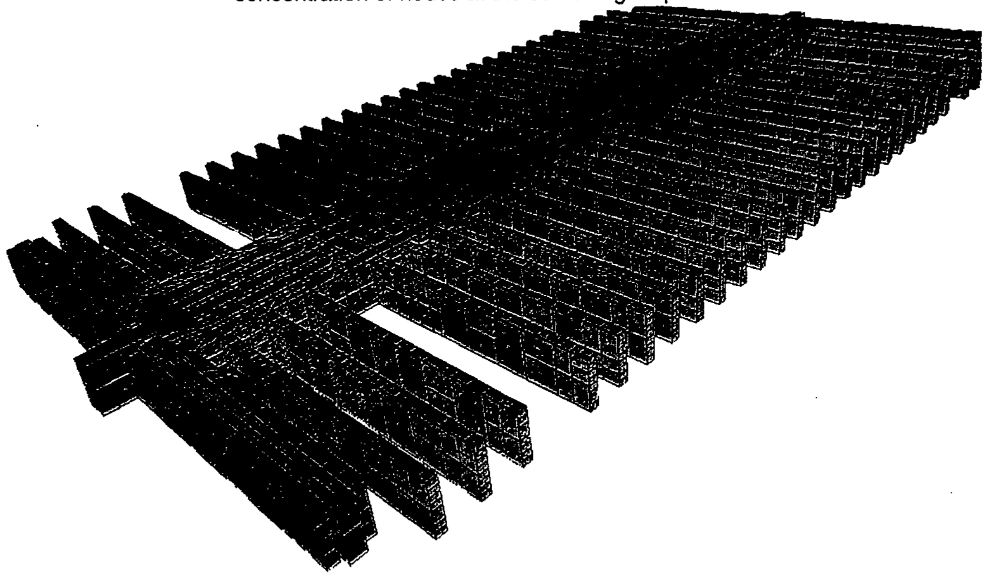
This is a very economical method to refine the edges and corners of structures. Electrostatic fields typically have singularities on corners which will be detected and refined using Section 5.8.4 "Error Indicator", but the error indicators will require multiple field solves to obtain the refined mesh.

5.8.7 Element Tunnel

All elements intersecting or enclosed by the specified tunnel are refined.

Tunnel criterion is very convenient for very long wires (or any wire like structures). From experience we know that most of the charge (over 99%) will be concentrated close to the critical net. Therefore we can relatively safely neglect the geometry that is outside of the area that is "close" to the critical net; that area is what we call the "tunnel". In practice this yields significant savings in memory and run times without compromising the solution. The part that is heuristic is the parameters "tunnel prune distance" and "tunnel distance". These will be different for different processes but a rule of thumb is to solve for one net with relatively large tunnel distance and than observe how much the result changes when you narrow the tunnel down. The pruning distance determines where the element will be chopped of if it falls outside the tunnel (if necessary). It should always be a little larger than the tunnel size. Our experience is that tunnel wide enough to capture only the nearest neighboring wires is typically good enough to yield sub 5% accuracy.

FIGURE 28. Detail of the adaptively refined ADXL76 moving proof mass. Note the automatic concentration of nodes at the comb-finger tips and anchors.



5.9 Adaptive Meshing Error Indicators

5.9.1 Initial Solution

BEM calculates very accurate solutions to PDEs. There are essentially only two sources of error in these calculations: (1) numerical integration accuracy and (2) geometric discretization accuracy. Coyote uses an automatic identification of the integration accuracy using Gaussian-Legendre integration to obtain fast and efficient numerical integration. The second item reflects that the model may have curved surfaces (approximated as a n-order polynomial) or that the state or gradient may vary nonlinearly (approximated as a n-order polynomial).

Currently Coyote only supports BEM elements with straight edges or planar faces. (The extensions to support curved edges or surfaces are straightforward, but require considerably more complex numerical and analytical singular integrations.)

To capture the effects of a nonlinearly varying solutions, the model discretization must allow an accurate "fit" to the physical solution. For example, if you know the result varies logarithmically, don't expect a good fit with a single linear element interpolation. However, a very good fit may be obtained using several linear elements or only a few quadratic elements.

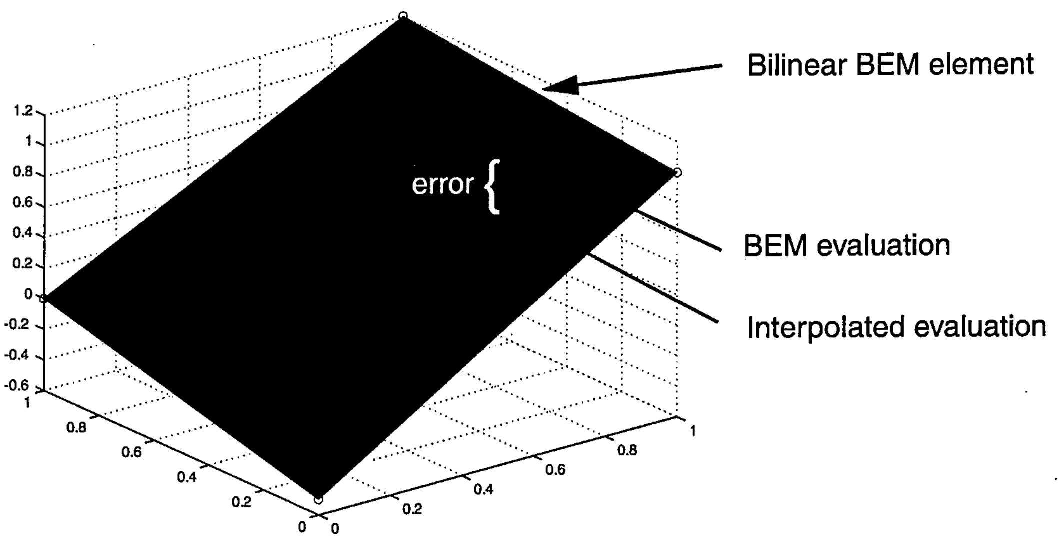
To ensure consistently accurate simulations, Coyote has developed an automatic method to mesh and refine the BEM model. This allows the user to specify a desired accuracy (e.g. 2% error on the states/gradients, or a 1% error on the capacitance) and then let the software refine the mesh until this accuracy is achieved.

To achieve this automatic mesh refinement, Coyote first solves the initial mesh and then uses post-processing to evaluate the states and gradients at points inside each element. For example, if a particular element is a triangle with a bilinear shape function, then the state/gradient error indicator evaluations on the surface should map closely to a plane face. If they do not, then this indicates that the size of the element should be reduced (h-refinement) or that the polynomial shape function of the element should be increased (p-refinement). Coyote uses both h-refinement and p-refinement to ensure accurate results. If all error indicators show that the elements are well-formed, then the adaptive refinement is finished and the results are reported to the user. If some elements have too large errors, then the model is resolved until all errors are below the user-specified limits.

5.9.2 Error Estimation

The errors for each element are obtained by comparing the interpolated BEM solution with new BEM point evaluations as shown in Figure 29.

FIGURE 29. BEM solution obtains exact values at the collocation nodes (vertices). Error indicators are obtained by comparing new independent BEM integral evaluations against interpolated shape function values.

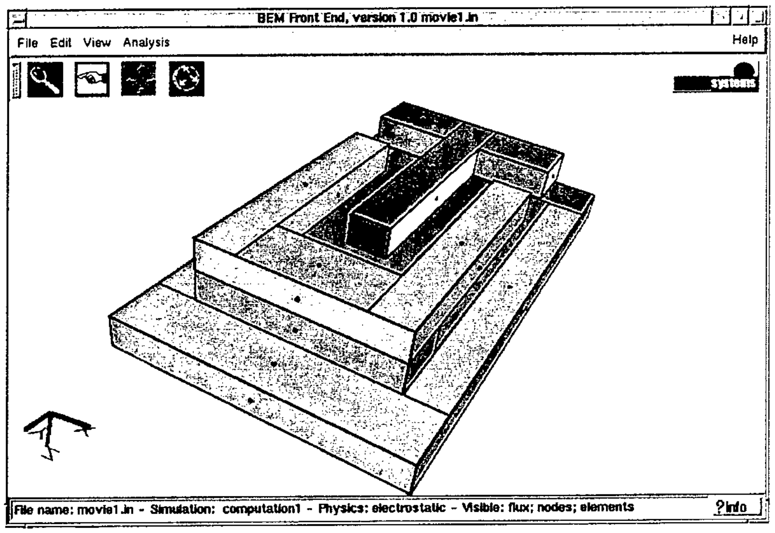


By determining the errors in the solution, the mesh can be locally refined using h-adaptively (panel size is decreased) and p-adaptively (shape function increases to linear, quadratic, cubic elements) as shown in Figure 32.

5.9.3 Example

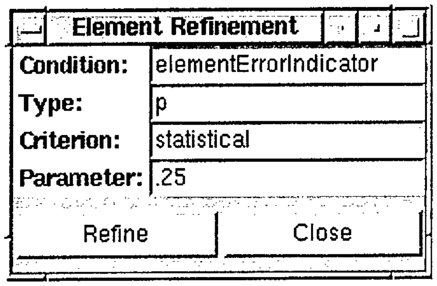
To illustrate this automatic adaptive mesh refinement, consider the initial BEM mesh of a comb-finger shown in Figure 30. Note that each element is either a triangular or quadrilateral with a single red dot. The number of red dots indicate the order of the shape function. A single dot indicates that the element is a constant element, 4 dots indicate a bilinear shape function, 9 dots indicate a biquadratic shape function and 16 dots indicate a bicubic shape function.

FIGURE 30. Comb-finger model using very coarse geometric discretization. A red dot indicates a degree of freedom. A single red dot in a BEM panel indicates a constant element. The colors indicate the magnitude of the solved electric field.



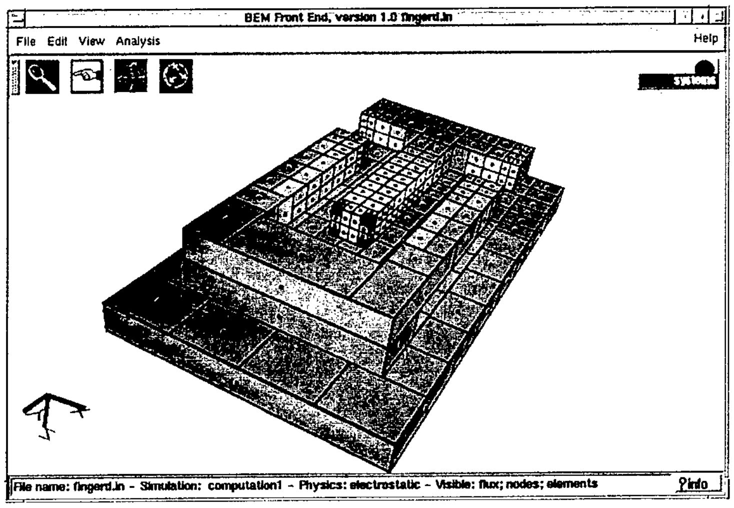
File name: movie1.in - Simulation: computation1 - Physics: electrostatic - Visible: flux; nodes; elements

FIGURE 31. Refinement GUI window



Condition:	elementErrorIndicator
Type:	p
Criterion:	statistical
Parameter:	.25

FIGURE 32. Adaptively refined mesh of comb-finger. Note the concentration of nodes at the tips and edges of the structures, where physically there is a singularity in the electric field. This automatic model refinement eliminates the need for an expert simulation user.



File name: fingerd.in - Simulation: computation1 - Physics: electrostatic - Visible: flux; nodes; elements

5.10 Volume-Based Forces

Coyote extended the software Generation 1 to support the Dual Reciprocity Boundary Element Method (DRBEM) to solve elastostatic gravitational/acceleration problems. This is useful to simulate the effect of acceleration on the static deformation of accelerometers. In contrast to electrostatic forces which act on the surfaces of models, gravity acts on the enclosed volume. Traditional BEM supports the surface-based forces but not the volume-based forces. However, the dual reciprocity extension to BEM allows volume forces to be expressed as radial functions of the surface nodes.

For example, the analytical deflection of a hanging beam fixed in Z along one face is given by

$$\begin{aligned} u &= -\nu\rho g x z/E \\ v &= -\nu\rho g y z/E \\ w &= \frac{\rho g z^2}{2\cdot E} + \frac{\nu\rho g}{2\cdot E}(x^2+y^2) - \frac{\rho g l^2}{2\cdot E} \end{aligned} \qquad \textbf{(EQ 1)}$$

where u, v, w are the deflections in the x, y and z directions, ρ is the material density, g is the gravitational acceleration, l is the beam length along the z-axis, ν is the poisson's ratio, and E is the material modulus of elasticity.

The DRBEM model of this beam and the deflected solution are shown in Figure 33. The excellent agreement of the DRBEM and analytical deflections are shown in Figure 34.

DRBEM has not been implemented in software Generation 2 or 3 since the DRBEM relies on manipulation of the direct BEM matrices which do not explicitly exist in the matrix-free FBEM implementation used in software Generation 2 and 3. As a result, note that the computational scaling of DRBEM is worse than $O(N^3)$ which entails that solving large models is computationally prohibitive.

Interesting future work could try to implement DRBEM in a matrix-free approach and utilize multipole acceleration to achieve excellent computational scaling.

FIGURE 33. Deflection of beam under gravitational load. The top end narrows and the bottom end widens due to the gravitational (volumetric) forces acting on the beam. (Isoparametric quadratic elements)

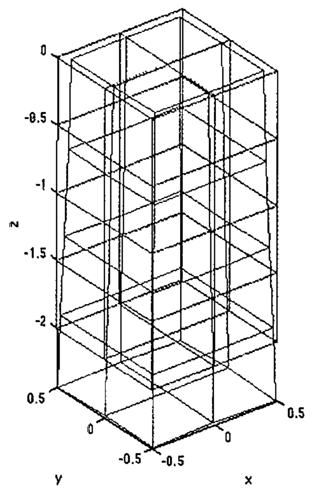
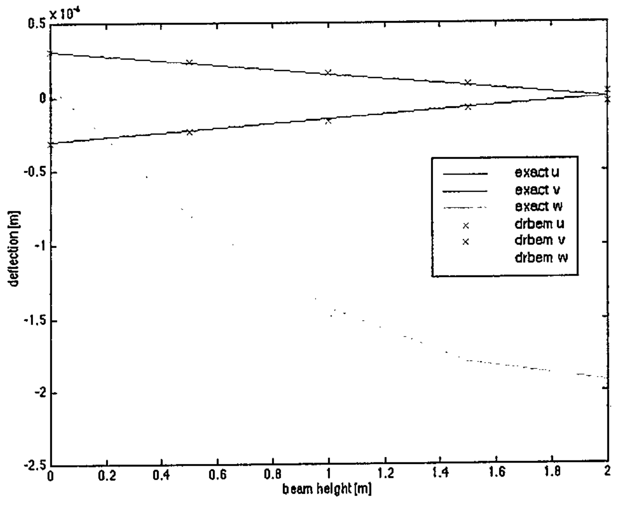


FIGURE 34. Excellent agreement of analytical and DRBEM numerical deflections due to volume forces arising from acceleration using isoparametric quadratic BEM elements



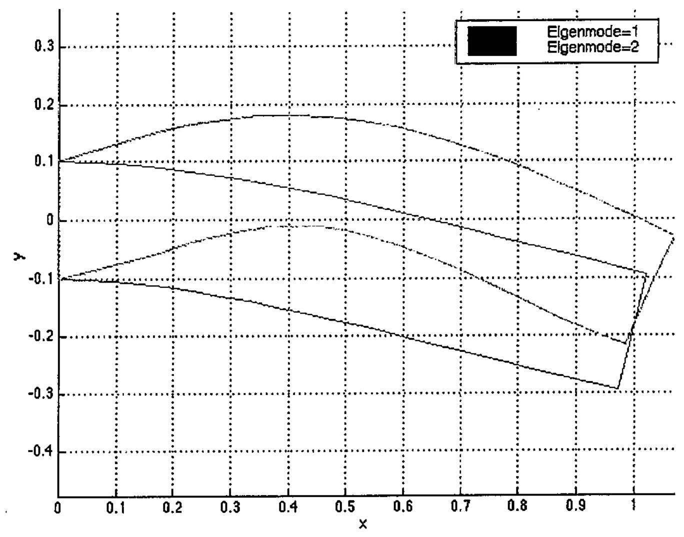
5.11 Modal Analysis

Coyote software generation #1 utilized the Dual Reciprocity Boundary Element Method (DRBEM) to solve modal analysis problems. In contrast to electrostatic problems which are Laplace problems, modal analysis cannot be solved using tradition BEM. However, the dual reciprocity extension to BEM allows modal analysis terms to be expressed as radial functions of the surface nodes.

Initial modal analysis results reveal that technical problems exist, and modal analysis capability could not be included in the software generation #1 release. Coyote worked with Prof. Partridge, the inventor of DR-BEM, in an attempt to resolve these technical issues. Unfortunately Partridge has been unable to provide Coyote any sample software code that correctly calculates modal analysis using DR-BEM as described in his technical papers[5].

5. P. Partridge, The Dual Reciprocity Boundary Element Method, Elsevier, 1992

FIGURE 35. Initial work: First 2 eigenmodes of a clamped-free beam

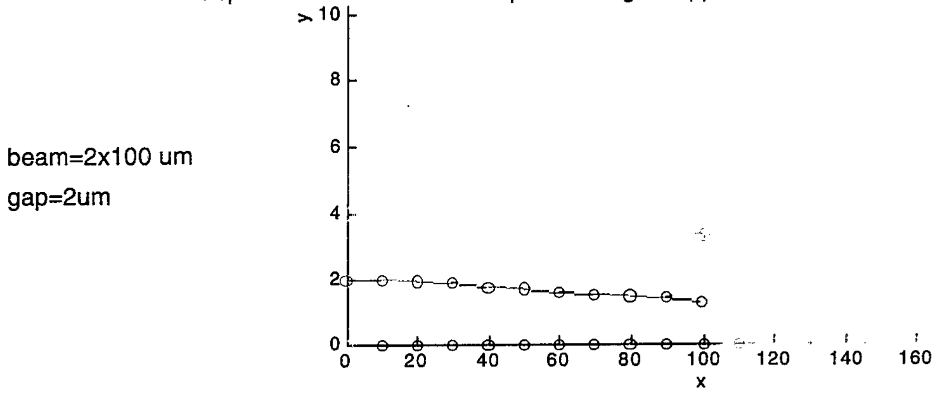


5.12 Coupled Electro/Elasto Simulations

MEMS simulations involves the interaction of electrostatic and elastostatic simulations. Coyote has chosen to simulate these effects using an electrostatic field solver and an elastostatic field solver coupled together using a relaxation algorithm.

As a simple example, consider the 2D cantilever beam suspended over an electrode shown in Figure 36. Note the simple BEM discretization of the boundaries. The colors indicate the solved electrostatic charge density on the converged model.

FIGURE 36. Typical MEMS electromechanical coupled problem consisting of a fixed-free beam over a groundplane. The color indicates the magnitude of the charge in the self-consistent displacement when 80% of the pull-in voltage is applied.



5.13 Relaxation Coupling

Relaxation is a simple iterative algorithm where the output results of one simulator are applied as inputs to the next simulator. For MEMS, the electostatic forces are applied as loading to the elastostatic model and the elastostatic simulator obtains the new deflections and stresses. The deflections are applied to the electrostatic geometry and the electrostatic field resolved to obtain updated electrostatic forces. This process is iterated until convergence, typically when the deflection or charges do not change appreciably.

5.14 Multi-level Newton Coupling

In addition to relaxation iteration, Coyote also developed a multi-level Newton iteration and implemented this in software release #1.

Published work by other groups indicate that the Newton method has a more distinct convergence near pull-in compared to simple relaxation iteration. Coyote assumes that MEMS self-consistent electro/mechanical simulations have converged when displacements do not change by more than a nanometer and capacitances do not change by more than an attofarad. Coyote was not able to create a realistic MEMS structure geometry and boundary conditions which exhibited poor convergence near pull-in, reducing the need for the Newton method. As a result, the simpler relaxation iteration appears to be perfectly acceptable. In almost all cases, realistic MEMS simulations converged in less than 10 iterations using relaxation iteration. The multi-level Newton method resulted in similar self-consistent results compared to the simpler relaxation algorithm, but at a significantly higher computational cost.

The Newton method uses multiple electro and multiple elasto field solves for each iteration, whereas Coyote only uses multiple electro and a single elasto field solve. It may be possible to combine the two coupling algorithms, where Coyote's method is used to rapidly approach the final solution, and then the Newton method is used to converge to the final solution if convergence problems are detected.

5.15 Efficient Coupling

Assuming that geometric nonlinearities are insignificant, it is not necessary to resolve the mechanical field at every iteration. Geometric nonlinearities arise in statically indeterminant clamped-clamped structures where the spring constant varies as a function of displacement (e.g. "stiffening"), whereas geometric nonlinearities are typically not found in clamped-free structures.

Instead the mechanical BEM matrices can be precomputed, and the electrostatic loading is applied as a new boundary condition. This dramatically reduces the mechanical cputime from a field solve to a linear matrix solve. Coyote calls this method "single elasto / multi electro" coupling. If the user is unsure whether geometric nonlinearities are significant, then the mechanical field solve must be recomputed at each iteration.

Table 5 shows estimates of coupled simulations contrasting BEM and CBEM and "multi elasto / multi electro" and "single elasto / multi electro" coupling. Since very little cputime is taken by the mechanical solver, it is not necessary to develop a fast elastostatic solver.

TABLE 5. Coupled Field Speedup possible using alternative Loading and Methods. Using the "single elasto / multi electro" loading and CBEM enables the fastest simulations. All values are estimates.

Error	Loading	Method	Electrostatic			Elastostatic			Coupled
			nodes	cputime	iteration	nodes	cputime	iteration	cputime
1%	multiple elasto/ multiple electro	BEM	30k	60 min	10	3k	60 min	10^*	1200 min
1%	single elasto/ multiple electro	BEM	30k	60 min	10	3k	60 min	1	660 min
1%	single elasto/ multiple electro	CBEM	4k	15 min	10	1k	15 min	1	165 min
10%	multiple elasto/ multiple electro	BEM	5k	10 min	10	500	10 min	10^*	200 min
10%	single elasto/ multiple electro	BEM	5k	10 min	10	500	10 min	1	110 min
10%	single elasto/ multiple electro	CBEM	1k	4 min	10	100	4 min	1	44 min

5.16 Problems with Elastostatic Multipole Operators

Coyote Systems has demonstrated a prototype fast BEM engine solving large, realistic MEMS electrostatic problems. To solve large, realistic MEMS elastostatic problems, Coyote originally thought it necessary that the elastostatic BEM calculations are also accelerated using a multipole algorithm.

As previously described, the BEM equations to simulate the Navier fields were not sparsified using multipole acceleration or similar technique. As a result of the O(N^3) scaling, only small models can be simulated before the computational requirements become prohibitive. It would be very interesting to create multipole acceleration operators for the Navier field equation. Another alternative is to implement precorrected FFT-acceleration which would be field equation independent.

Coyote is only aware of 2D elastostatic multipole operators. Unfortunately, the multipole coefficients are not defined in a hierarchical manner, where the coefficients of parent multipole clusters depend only on the coefficients of child multipole clusters. As a result, there is an extremely significant cputime cost to calculate the multiple coefficients. Further, the techniques used to develop 2D multipoles are not suitable for 3D multipoles. Coyote investigated multipole operators for elastostatics using Mathematica but did not implement these in the Coyote cad software.

As a result of these reasons, Coyote has investigated alternate methods (See "Constrained BEM" on page 36., See "Tunnel Acceleration" on page 38., See "Relaxation Coupling" on page 49.) to ensure fast coupled simulations.

5.17 Problems with Mechanical Body Forces

The Boundary Element Method uses surface based models and reformulated partial differential equations expressed on the surfaces. As a result, the models are typically very small since only the surfaces are discretized, while the interior volumes are not.

BEM is very suited for simulations involving boundary or surface forces, such as electrostatics. However, it is not well suited for body forces acting on the volume, such as gravity. The Dual Reciprocity extension to the Boundary Element Method (DR-BEM) offers a method to allow arbitrary forces to be expressed as surface forces, allowing successful simulation without discretizing the volume of the model.

Coyote successfully demonstrated the use of DR-BEM to solve for gravity and acceleration on simple structures using the first generation BEM software. DR-BEM involves matrix manipulation of the naive BEM matrices shown in Eq. 3 on page 31.

Coyote did not implement DR-BEM in second or third generation BEM software, since these later versions were accelerated using matrix-free algorithms. As a result, it would have been expensive to explicitly calculate the full matrices to allow DR-BEM. Coyote has therefore not demonstrated acceleration on large, realistic MEMS devices

5.18 Problems with Modal Analysis

Coyote attempted to solve modal analysis using the Dual Reciprocity extension to the Boundary Element Method (DR-BEM). Unfortunately we could not successfully replicate modal analysis experiments performed by Dr. Partridge, a noted author on DR-BEM. Subsequent contact with Dr. Partridge regarding these problems showed that his simulations did not correctly simulate the elastostatic modal analysis, but instead used a much simpler mathematical Helmholtz equation.

An extensive literature search shows that modal analysis using BEM has not been conclusively demonstrated.

5.19 Summary

A surface-based numerical field solver using multipole accelerated BEM has shown itself to be very useful for simulating large realistic MEMS devices. A relaxation algorithm between separate electrostatic and elastostatic simulations has been demonstrated to be simple and efficient. Acceleration of elastostatics would be beneficial for improved speed and the capability to simulate even larger mechanical models.

6.0 Software Releases

The software architecture, development environment, verification and quality control are all elements of providing the users with a stable, well performing CAD system. Three software releases were made with varying capabilities. The desired capabilities of CAD software for MEMS changed slightly through the various software generations. Early releases featured many different technical capabilities but could only solve small problems. Later releases emphasized fast simulations of complex geometrical models.

The final software Generation 3 is shipping under the registered trademark AutoMEMS, and provides the user with an easy-to-use system to generate meshed 3D models directly from 2D layout masks and to simulate coupled electro-mechanical MEMS devices. Tutorials guide the beginner through simple exercises culminating in simulations of the Analog Devices ADXL76 MEMS accelerometer and Texas Instruments MEMS micromirror.

6.1 Development System

Initially software Generation 1 utilized the Matlab environment which allowed a portable executable, data format and graphics environment. However the large size of the BEM engine made support and development under Matlab cumbersome. For example the debugging capabilities of Matlab are rudimentary, and the non-standard object oriented features are primitive.

As a result of the Generation 1 experience using Matlab, Generation 2 software was developed using C++ with standard gnu-style compilers and tools.

Linux was chosen as a "plain vanilla" unix making it is quite simple to port to other unix flavors (e.g. IBM, Sun, Digital, HP). Coyote developed the software under the Redhat Linux operating system on Intel PC hardware. This is an economical and high-performance solution.

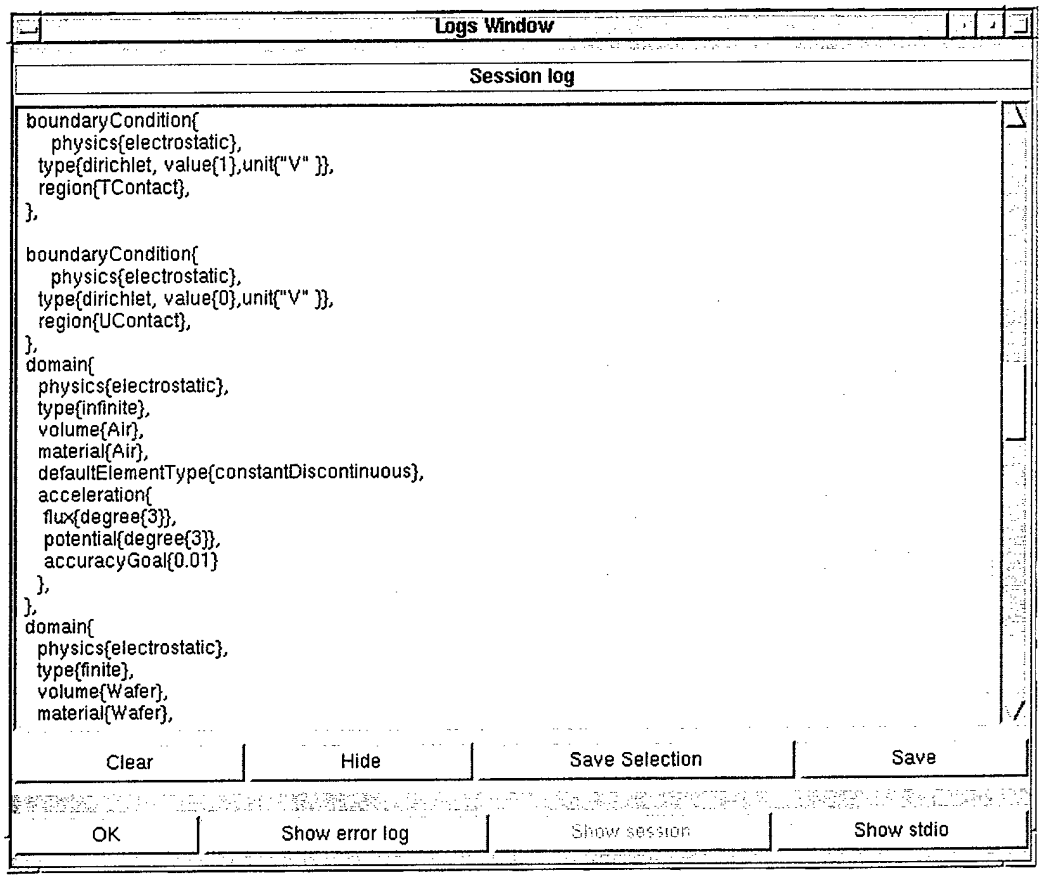
6.2 Client/Server Architecture

Coyote's software is written in a modern object-oriented client/server architecture. The client and server communicate over a TCP/IP socket interface, which allows the client (e.g. GUI) and server (e.g. BEM engine) to run on separate machines and communicate over the internet. Because the BEM models are very small, the communication overhead is minimal.

The client and server communicate using a human-readable hierarchical command language documented on Coyote's website. An example taken from the log of the GUI is shown in Figure 37.

Coyote developed a node-locked license manager and encrypted license key system. This enables Coyote to distributed time-limited and/or limited functionality software versions.

FIGURE 37. Session log



```
boundaryCondition{
    physics{electrostatic},
  type{dirichlet, value{1},unit{"V" }},
  region{TContact},
},

boundaryCondition{
    physics{electrostatic},
  type{dirichlet, value{0},unit{"V" }},
  region{UContact},
},
domain{
  physics{electrostatic},
  type{infinite},
  volume{Air},
  material{Air},
  defaultElementType{constantDiscontinuous},
  acceleration{
   flux{degree{3}},
   potential{degree{3}},
   accuracyGoal{0.01}
  },
},
domain{
  physics{electrostatic},
  type{finite},
  volume{Wafer},
  material{Wafer},
```

6.3 Software Releases

Although the real value to the customer is the high-speed high-accuracy C++ BEM engine, customer satisfaction is also tied to ease-of-use, look and feel, and portability. Using team members as Beta sites, we will obtain feedback on what the user needs and desires are for the look and feel for the tool as well as provide to them the powerful BEM engine for their evaluation.

6.3.1 Generation 1

Generation 1 of the software Coyote created for coupled electrostatic-mechanical analysis of MEMS using boundary element software (BEM) was created in the Matlab environment. Matlab was an excellent prototyping environment featuring portable software data formats and graphics. Later versions of the Generation 1 BEM engine became quite complex consisting of several hundred Matlab *.m files. As a result the debugging environment was ineffective. More significantly, the poor computational performance of the

Matlab system dramatically reduced the size of models that the Generation 1 software could simulate.

As a result of these limitations, the second generation software would be created in a C++ environment with gnu-style tools and high-speed portable OpenGL graphics.

6.3.2 Generation 2

Generation 2, released in 98Q3, has demonstrated the power and accuracy of an accelerated BEM engine. Generation 2 is capable of solving very large electrostatic systems up to 250,000 BEM nodes. As shown in Section 2.3 on page 8, 200,000 BEM nodes is equivalent to approximately 100,000,000 FEM nodes. Note that 10,000,000 FEM nodes is as large as the largest FEM problem solvable today. Using only 50,000 BEM nodes, an entire device can be simulated as shown in Figure 16 on page 26. No other application has demonstrated this capability.

Generation 2 solves large electrostatic problems because of three major features: adaptive meshing, multipole acceleration, and constrained elements. Adaptive meshing permits the automatic refinement on areas of the model that will have the greatest change in state and/or gradients enabling more accurate results. Multipole acceleration enables fast simulations by clustering distant elements together, resulting in a single value for that cluster. Constrained BEM allows local refinement for high accuracy without the mesh refinement propagating to other parts of the model.

6.3.3 Generation 3

The final software release in 99Q3 added coupled simulation capability to simulate large, realistic coupled MEMS devices in a timely manner. This software utilized the C++ framework developed in Generation 2.

6.3.4 Capabilities of software

The capabilities of the Generation 1, 2 and 3 software are summarized in Table 6. The Generation 3 software is fully capable of simulating large complex MEMS devices like the Analog Devices ADXL76 or Texas Instruments micro-mirrors on standard workstations. Typical high-speed high-accuracy simulations, which include generating a meshed 3D model from 2D layout masks, are completed within several minutes of cpu-time.

TABLE 6. Software capability summary by software generation

Item		Gen 1	Gen 2	Gen 3
Platform	Windows NT	X		X
	Unix	X	X	X
	Mac	X		
	Portable Executable	X		
	Portable Source	X	X	X
GUI		MATLAB	tcl/tk	tcl/tk
BEM Engine		MATLAB	C++	C++
Visualization		MATLAB	vtk	vtk
Features	Solid Model			X
	Automatic Mesh		X	X
	H-type Adaptive Mesh		X	X
	P-type Adaptive Mesh		X	X
	2D Electrostatic	3,000 Nodes	1M+ Nodes	not tested
	3D Electrostatic	3,000 Nodes	250,000 Nodes	2M+ Nodes
	2D Elastostatic	300 Nodes	-	not tested
	3D Elastostatic	300 Nodes	-	not tested
	Accelerated Electro		X	X
	Coupled Solver			X
	Multi-Physics	X		X
	Higher Order Elems		X	X
	CAD Import/Export			X

Note that each electrostatic 3D BEM coefficient is a scalar, while elastostatic 3D BEM coefficients are 3x3 tensors. Simulating a 3D elastostatic field equation is therefore approximately 9x more expensive than a 3D electrostatic model. This explains why the Generation 1 software can solve 300 elastostatic nodes and 3000 electrostatic nodes. The prototype of Generation 2 software has demonstrated solving 250,000 electrostatic nodes (which would correspond to about 25,000 elastostatic nodes).

6.4 Quality Control & Testing

A commercial development environment was used to maintain "check-in" and "check-out" privileges for code under development. All submitted code was rigorously tested by automated test suites every evening. All encountered bugs resulted in a test program being added to the test-suite to ensure that problem was not inadvertently repeated.

All code is exercised using the API format that Coyote uses for all client-server applications. In this case, instead of using a GUI to create the API calls, the test-suite program generates the API calls to fully exercise the engine.

Coyote has documented the developed software in a 400+ page manual available online. The documentation includes descriptions of the Coyote file formats, API command language, MEMS tutorials and verification examples. Examples of some of the verification cases follow below.

6.4.1 Electrostatic Verification

Several 2D and 3D geometries with multiple materials were used to compare the Coyote electrostatic simulations with analytical results. All verification models (which are included in the documentation available online at http://www.coyotesystems.com/pdf/autobem.technical.pdf) demonstrated that the BEM software was extremely accurate.

An example of a verification model are the concentric spheres shown in Figure 10.

All models with analytical results have a fairly simple topology (e.g. cylinders, spheres, cubes) to allow an analytical solution to be derived. As a result, the simulations of models with complex topologies cannot be verified. However we have not identified any problems simulating complex geometries. Comparision of Coyote numerical simulation results with actual measurements of the Analog Devices ADXL76 MEMS accelerometer show excellent agreement with <2% errors.

6.4.2 Elastostatic Verification

Coyote has implemented and verified several 2D elastostatic verification problems as seen in Figure 38 and Figure 39. These verification cases where chosen because the simple topologies and boundary conditions allow analytical solutions to be derived. All verification models (which are included in the documentation available online at http://www.coyotesystems.com/pdf/autobem.technical.pdf) demonstrated that the BEM software was extremely accurate.

The 2D elastostatic verifications agreed extremely well with the analytic solutions as shown in Figure 38 and Figure 39. Unfortunately no prismatic 3D analytic models could be located in any references that were suitable as verification cases.

FIGURE 38. Elastostatic 2D clamped-free beam under constant distributed load on top surface. Note the good agreement with the analytical solution.

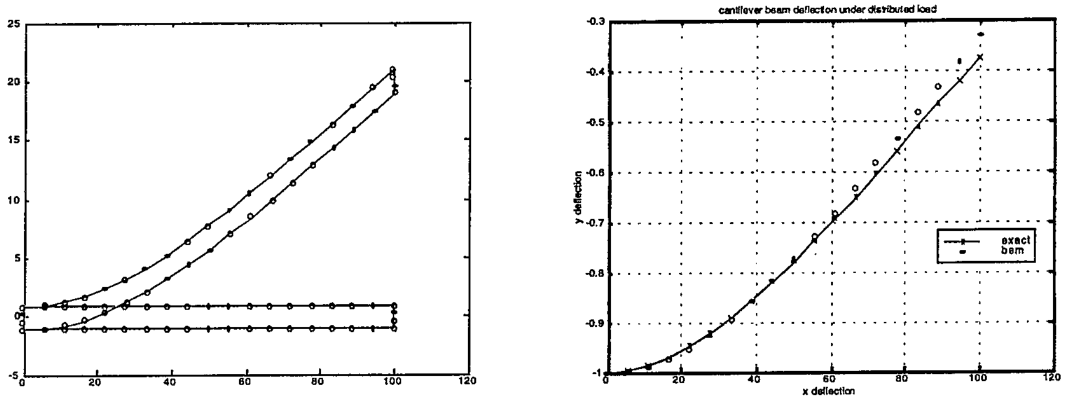
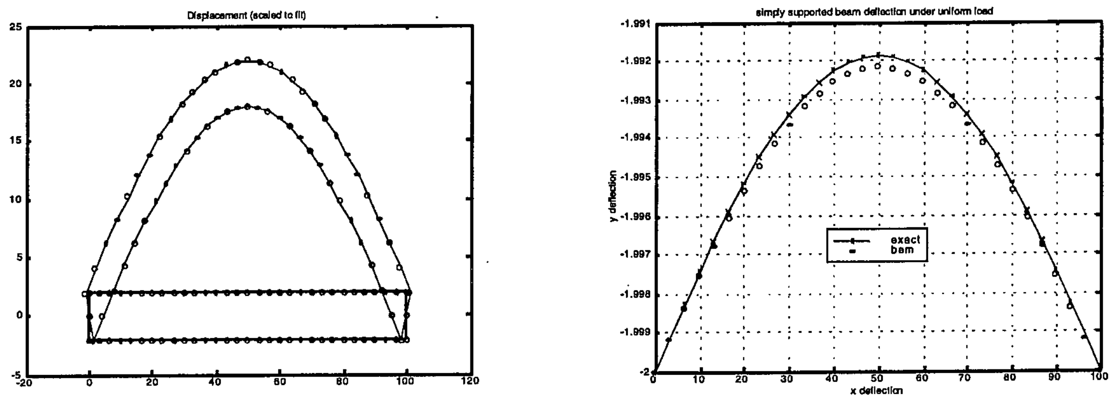


FIGURE 39. Elastostatic 2D pinned-pinned beam under constant distributed force. Note the good agreement with the analytical solution.



6.4.3 Coupled Electro-mechanical Verification

We are not aware of any 3D coupled problem that can be used as a simple verification problem. However, we used several simple 2D problems to ascertain that the coupling was working satisfactorily.

6.5 Problems with Matlab

Coyote's first generation BEM software was written in the Matlab language, which was a wonderful prototype environment offering fast matrix manipulations and cross-platform graphics. The first generation tool consisted of several hundred *.m files and functions, and Matlab officials commented on that this was probably one of the largest Matlab applications they had encountered.

To continue to add functionality, data abstractions, and debugging capability to Matlab proved extremely challenging. Further, the performance of Matlab on large matrices was quite poor. In addition, the object oriented (OO) features in Matlab v5 are immature and incomplete.

As a result, Coyote preferred to recode the second and third generation software using multipole accelerated BEM engine in C++.

6.6 Summary

The high-quality and performance of the software releases to Beta users has demonstrated that the architecture, development environment and testing has been successful. No Beta user has been able to identify a "bug" that the development team was not already aware of. Some Betas have been able to suggest a clearer or alternative simulation strategy which have been incorporated into the final AutoMEMS software.

7.0 Interaction with Composite CAD teams

The team that developed the capabilities of the AutoMEMS software through Generations 1, 2 and 3 included Coyote Systems, UC Berkeley, Texas Instruments, Hewlett-Packard, and Sandia. The groups other than Coyote were chosen as typical MEMS users and their input would be used to develop a stronger more robust and capable software tool. Unfortunately these team members did not participate to the extent desired. Analog Devices became a strong proponent of the AutoMEMS software and their inputs and modeling challenges strongly influenced the capabilities of the AutoMEMS software as released in the final Generation 3 release.

Collaboration with other groups was made during the DARPA Composite CAD program. In particular Coyote wanted to work with groups that had a strong model generation capability since originally this was not part of the Coyote tool. For this reason Coyote worked with Tanner and MemsCap. Some capabilities were demonstrated, but no vendor was able to demonstrate acceptable model generation capabilities. As a result, Coyote developed its own acceptable model generation software included as part of AutoMEMS. Coyote also worked with Ptolemy-group at UC Berkeley to demonstrate high level simulation capabilities utilizing the Coyote field solver.

7.1 Analog Devices

"AutoMEMS represents a significant advancement in the state-of-the-art for coupled electromechanical simulation. In an afternoon, we were able to download the software, create a meshed 3D model directly from our accelerometer layout, simulate it, and verify that the results were within 5% of our measured data" said Nicholas Swart, MEMS CAD Manager at Analog Devices. "AutoMEMS is a valuable component in our MEMS design flow."

Analog Devices has provided the 2D masks for the ADXL76 accelerometer. Coyote successfully created a 3D model of the accelerometer from the masks (including etch-holes). By utilizing adaptive meshing, an accurate electrostatic simulation was obtained with approximately 50,000 BEM panels. This simulation took less than 1 minute on a PC.

FIGURE 40. The 3D model of ADXL76 formed from the masks is meshed for CBEM.

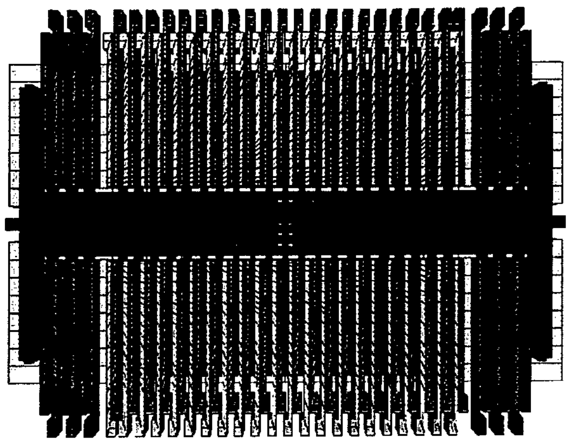


FIGURE 41. Detail of the adaptively refined XL50 moving proof mass. Note the automatic concentration of nodes at the comb-finger tips and anchors.

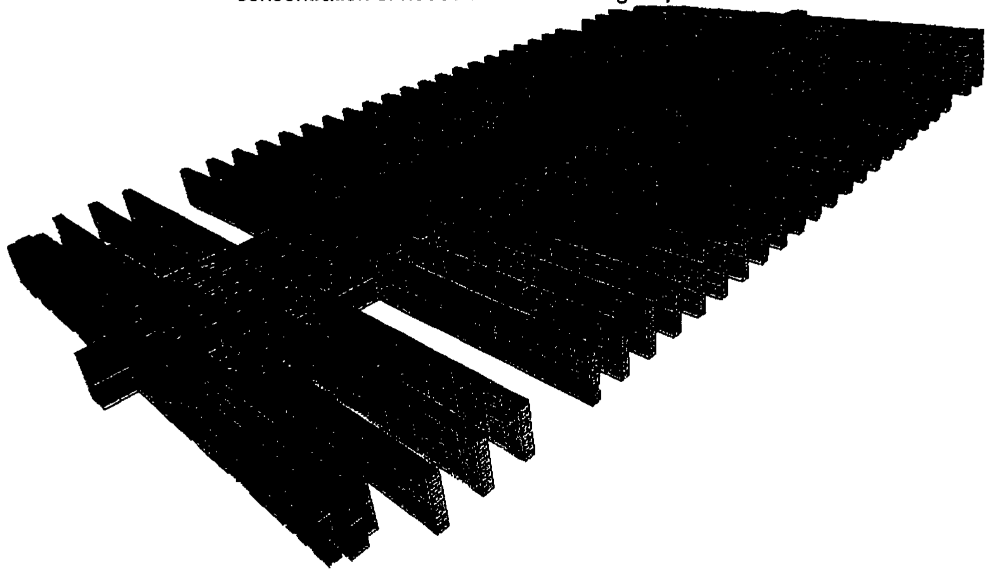
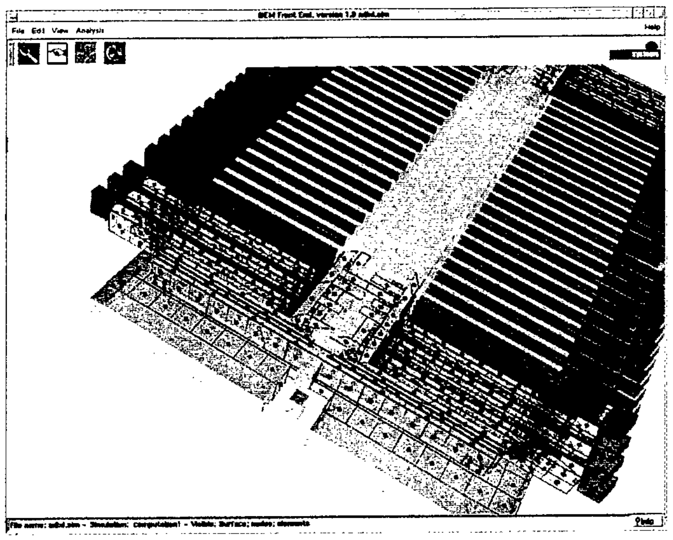


FIGURE 42. Tunnel around magenta/yellow driving comb-fingers in ADXL76. The blue/purple sense comb-fingers are not included in the tunnel making the BEM model significantly smaller.

Instead of solving the entire XL50 in 20 min, this tunneled BEM problem is solved in 90 sec, which is a 40x savings in memory and cputime.



7.2 Tanner

Technical information regarding Coyote's products and capabilities has been shared with Tanner. Tanner has demonstrated that prototype Tanner MemsPro tools could create a valid Coyote inputdeck for analysis by Coyote software generation #3. Tanner utilized the ACIS geometry engine inside the MemsPro tools and could therefore generate a meshed ACIS model that AutoMEMS could import.

FIGURE 43. Coyote analysis of a Tanner supplied Coyote-formatted inputdeck demonstrating interoperability

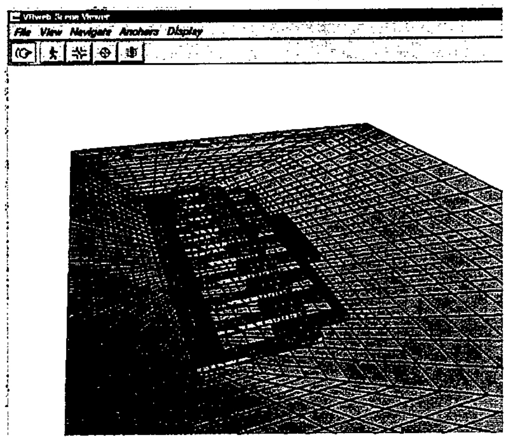
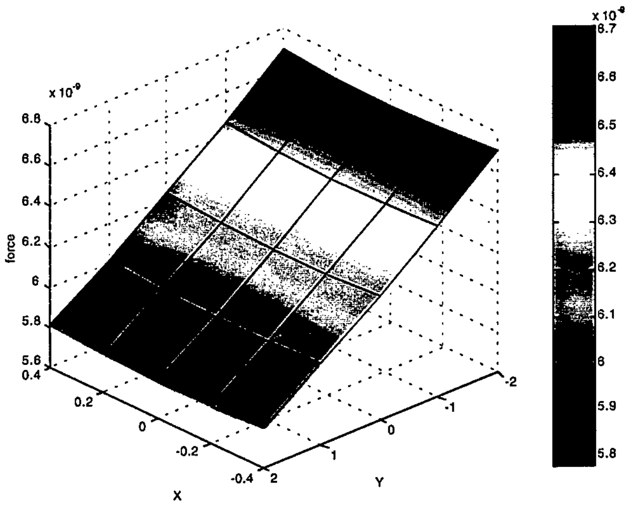


FIGURE 44. The Coyote calculated charge, capacitance and forces for various resonator deflections was calculated and provided to Tanner demonstrating macromodel interoperability



7.3 UC Berkeley - Ptolemy

Technical information was shared with the UC Berkeley Ptolemy group to extend Ptolemy to solve PDEs using Coyote's BEM solver. Ptolemy is an academic software environment which allows multiple data formats and multiple data engines for advanced co-simulation. Unfortunately Ptolemy is not utilized by mainstream vendors.

Coyote did not continue working with the Ptolemy software and instead concentrated on generating HDL models for Simulink and Spice.

7.4 University of Illinois Urbana-Champaign

A technical liaison with Univ. Illinois was established with Prof. Narayan Aluru to determine whether the Newton multi-physics coupling was superior to the simpler relaxation iteration. A prototype of a coupled Newton-iteration has been implemented in Matlab.

As noted in Section 5.15 on page 50, Coyote chose to use relaxation since the dramatically increased computational costs of Newton coupling did not appear to be warranted.

7.5 MemsCap

Technical information regarding Coyote's products and capabilities has been shared with MemsCap. MemsCap offered to convert its FEM generator tools to BEM output, but this never occurred. As a result, no progress was made integrating the Coyote and MemsCap tools.

7.6 Summary

Collaboration with other groups was made during the DARPA Composite CAD program to demonstrate interoperability and demonstrate capabilities not offered by the Coyote AutoMEMS software. Unfortunately no other vendors provided robust tools that offered capabilities that Coyote had not previously demonstrated. One user group, Analog Devices, became a strong proponent for high-speed high-accuracy simulations possible with the AutoMEMS software and provided many large and complex MEMS models. Reports comparing the simulations with experimental data are being prepared.

8.0 Conclusions

Coyote Systems developed a high-speed high-accuracy coupled electro-mechanical simulator that successfully simulates large, realistic MEMS devices. This CAD system has been delivered to Coyote's Beta users under this program. Some noteworthy simulations are presented below, as well as recommendations for future work.

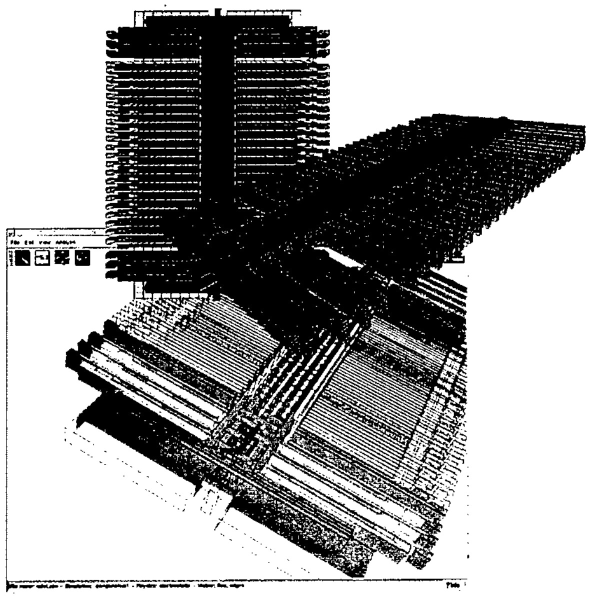
8.1 Demonstrated Successes

8.1.1 Analog Devices ADXL76 Accelerometer Electrostatic Simulation

Coyote has demonstrated the high-speed high-accuracy electrostatic simulation of the Analog Devices ADXL76 MEMS accelerometer. Starting from 2D layout, Coyote creates a meshed 3D model and simulates the electrostatic charges and forces in under 60 seconds on a 400 MHz PentiumIII computer. ADI reports excellent agreement with between the simulation and experimental results.

ADI further reports that no other simulator could simulate the entire device without extensive "geometric defeaturing", i.e. simplifying the layout removing combfingers and or etch-holes.

FIGURE 45. ADI76 simulation directly from layout



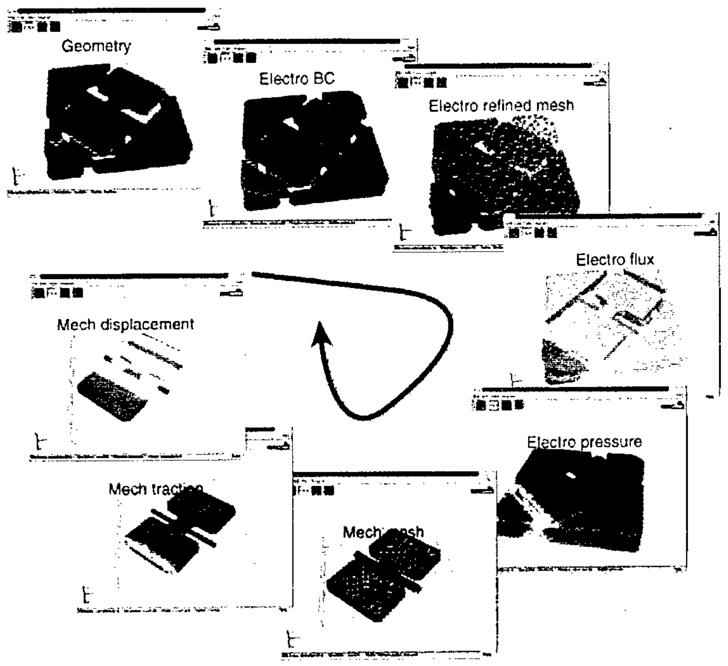
1. Import GDSII Layout
2. Automatic 3D Model Generation
3. Automatic BEM Mesh Refinement
4. High-Speed High-Accuracy 3D Field Solution
5. Accurate Capacitance & Force Extraction

8.1.2 TI Micromirror Coupled Simulation

Coyote has demonstrated the successful coupled electrostatic-elastostatic simulation of the Texas Instruments' MEMS micromirror. Starting with a layout description, a 3D meshed model was created, boundary conditions applied, and simulated until convergence.

TI reports that other tools grossly simplify the geometry of the micromirror to simulate it. Further, TI reports that all other tools require extensive human intervention to modify or correct other tools' meshing in order to apply boundary conditions. TI was very pleased that no such human-intervention by expert simulator users was required to use the Coyote tools.

FIGURE 46. Coupled simulation of TI micromirror



1. Import Geometry
2. Define Electro BC
3. Define Mech BC
4. Refine Electro Mesh
5. Solve Electro flux
6. Calc Electro pressure
7. Refine Mech mesh
8. Apply Mech traction
9. Solve Mech displacement
10. Done!

8.1.3 Coyote beats Supercomputer with PC

Simulating large models of PDEs with FEM is clearly impractical as shown by Table 1, "Comparison of Computational Scaling BEM vs. FEM," on page 8. For this reason numerical analysts at Sandia chose to implement Direct BEM on a large supercomputer, for which they won first prize in the 1994 IEEE Gordon Bell competition to advance high-performance computing. Sandia ran several direct boundary element method programs on an Intel Paragon supercomputer.

Today Coyote Systems' software is able to significantly improve on Sandia's supercomputer performance using a single personal computer.

A figure of merit that combines price, speed and model size demonstrates that Coyote Systems' BEM Engine achieves a 7 million times improvement over the supercomputer application. On news of the improvement, Dr. Gordon Bell, the creator of the IEEE Gordon Bell prize, called the work "Really exciting - better algorithms beat brute force!"

TABLE 7. Comparison of Coyote fast BEM and supercomputer BEM performance

	Sandia (1994) [a]	Coyote (1999) [b]	Comment [c]
Processor Nodes	1900x Intel i860	1x Intel PentiumIII	2000x cheaper
Maximum throughput	1.45E5 elements/hour	>1.25E6 elements/hour	9x faster
Maximum model size	50E3 elements	>2E6 elements	400x larger
Combined FOM			7E6x improvement

a. http://www.cs.sandia.gov/HPCCIT/g_bell_p2.html
b. http://www.coyotesystems.com/pdf/CMES.pdf
c. not running same models or simulations

8.2 Recommendations for Future Work

Most of the objectives made by Coyote have been successfully demonstrated and delivered to Beta users. Reflecting on users' success using the AutoMEMS software, it is natural for ideas on improving the existing tool to arise.

Some ideas involve improving on analytical capability in order to address modal analysis, fast BEM for elastostatics, and thermoelastic behavior. Additional work could also focus on facilitating the integration of AutoMEMS results with other macromodeling efforts. Finally, it would benefit AutoMEMS users to pursue additional ways of generating 3D models and importing MCAD models.

Appendix A. Publications & Conferences

The following papers were published in 1997 - 1999 by Coyote personnel. These are all available online[1].

A.1 1997

1. M. Bächtold, M. Emmenegger, S. Taschini, J. Funk, K. Lamboglia, R. Rühl, J.G. Korvink, H. Baltes, "A CAD System for Microsystem Simulation", Proc. MINAST 1st Ann. Conv., Berne, 1997
2. M. Bächtold, J.G. Korvink, H. Baltes, "The Adaptive, Multipole-Accelerated BEM for the Computation of Electrostatic Forces", Proc. CAD for MEMS, Zurich, 1997
3. P. Ljung, "Sequentially Solving Coupled Field Equations using BEM", Proc. CAD for MEMS, Zurich, 1997
4. M. Bächtold, M. Emmenegger, J.G. Korvink, H. Baltes, "An Error Indicator and Automatic Adaptive Meshing for Electrostatic Boundary Element Simulations", accepted for publication in IEEE Trans. on CAD of Integrated Circuits and Systems, 1997
5. M. Bächtold, S. Taschini, J.G. Korvink, H. Baltes, "New Convergence Scheme for Self-Consistent Electromechanical Analysis of iMEMS Automated Extraction of Capacitances and Electrostatic Forces in MEMS and ULSI Interconnects from the Mask Layout", Proc. IEDM, IEEE, 1997
6. M. Bächtold, J.G. Korvink. H. Baltes, "An Error Indicator and Automatic Adaptive Meshing for 3D Electrostatic Boundary Element Simulations", BEM 19, Rome Italy, Sept 9-12 1997
7. P. Ljung, "Solving PDEs with the Matlab Boundary Element Toolbox", Matlab 97 Conference, San Jose, CA, Oct 6-8 1997

A.2 1998

8. M. Bächtold, P. Ljung, "The Constrained Boundary Element Method: A Technique for Allowing General Surface Meshes in MEMS Simulations", accepted IEEE workshop at Hilton Head Island, SC June 1998
9. P. L. Levin, M. Spasojevic and R. Schneider, "Creation of Sparse Boundary Element Matrices for 2-D and Axi-symmetric Electrostatics Problems Using the Bi-Orthogonal Haar Wavelet" accepted IEEE Transactions on Dielectrics and Electrical Insulation, vol. 5, 1998
10. M. Bächtold, "An Error Indicator and Automatic Adaptive Meshing for Electrostatic Boundary Element Simulations" accepted IEEE CAD of Integrated Circuits and Systems 1998
11. P. Ljung, "An Automatic Method for MEMS Analysis", accepted Symposium on Computational Methods for Analysis, Design and Simulation of MEMS, ASME, Anaheim, Nov. 15-20, 1998

1. http://www.coyotesystems.com/pdf

12. P. Ljung, "Efficient MEMS CAD Tools", invited contribution to Special Issue on MEMS CAD Tools, Journal of Micromachining, June 1998

13. Ch. Lage, "Concept oriented design of numerical software", Technical Report 98-7, Seminar of Applied Mathematics, ETH Zurich, Jan 1998

14. Ch. Lage and Ch. Schwab, "Wavelet Galerkin algorithms for boundary integral equations", SIAM J. Sci. Stat. Comput., 20(6):2195-2222, 1998

A.3 1999

15. Ch. Lage and S. A. Sauter, "Transformation of hypersingular integrals and black-box cubature", Math. Comp, to appear

16. M. Bächtold, "An Error Indicator and Automatic Adaptive Meshing for Electrostatic Boundary Element Simulations", IEEE CAD of Integrated Circuits and Systems, submitted

17. M. Bächtold, M. Spasojevic and P. Ljung, "A System for Full-Chip and Critical Net Parasitic Extraction for VLSI Interconnects using a Fast 3D Field Solver", IEEE Trans. CAD, to appear

18. P. Ljung, M. Bächtold and M. Spasojevic, "Analysis of Realistic, Large MEMS Devices", Computer Modeling in Engineering & Sciences, to appear

19. M. Bächtold, M. Spasojevic and P. Ljung, "A System for Full-Chip and Critical Net Parasitic Extraction for VLSI Interconnects using a Fast 3D Field Solver", Design Automation Conference 2000, to appear

DISTRIBUTION LIST

addresses	number of copies
CLARE D. THIEM AFRL/IFTC 26 ELECTRONICS PKWY ROME NY 13441-4514	15
DR. PER LJUNG COYOTE SYSTEMS 2740 VAN NESS AVENUE #210 SAN FRANCISCO CA 94109	2
AFRL/IFOIL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	1
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
DR ANANTHA KRISHNAN DARPA/MTO 3701 N FAIRFAX DRIVE ARLINGTON VA 22203	5

MISSION OF AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.